
UNIVERSIDAD CARLOS III DE MADRID

# TESIS DOCTORAL

## VISIBLE LIGHT AND DEVICE-TO-DEVICE COMMUNICATIONS: SYSTEM ANALYSIS AND IMPLEMENTATION

Autor: Qing Wang

Director: Domenico Giustiniano, Ph.D.

Tutor: Ruben Cuevas Rumin, Ph.D.

DEPARTAMENTO DE INGENIERÍA TELEMÁTICA

Leganés (Madrid), Marzo de 2016


UNIVERSIDAD CARLOS III DE MADRID

# PH.D. THESIS

## VISIBLE LIGHT AND DEVICE-TO-DEVICE COMMUNICATIONS: SYSTEM ANALYSIS AND IMPLEMENTATION

Author: Qing Wang

Director: Domenico Giustiniano, Ph.D.

Tutor: Ruben Cuevas Rumin, Ph.D.

DEPARTMENT OF TELEMATIC ENGINEERING

Leganes (Madrid), March 2016

*Visible Light and Device-to-Device Communications: System Analysis and Implementation*

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Prepared by
Qing Wang, IMDEA Networks Institute & University Carlos III of Madrid

Under the advice of
Domenico Giustiniano, Ph.D., IMDEA Networks Institute
Ruben Cuevas Rumin, Ph.D., University Carlos III of Madrid
Departamento de Ingeniería Telemática, Universidad Carlos III de Madrid

---

Date: Marzo, 2016

Web/contact: qing.wang@imdea.org

This work has been supported by IMDEA Networks Institute.

TESIS DOCTORAL

# VISIBLE LIGHT AND DEVICE-TO-DEVICE COMMUNICATIONS: SYSTEM ANALYSIS AND IMPLEMENTATION

Autor: Qing Wang
Director: Domenico Giustiniano, Ph.D.
Tutor: Ruben Cuevas Rumin, Ph.D.

Firma del tribunal calificador:

Presidente:

Vocal:

Secretario:

Calificación:

Leganés, de de

# Acknowledgements

Firstly, I would like to thank my advisors Dr. Domenico Giustiniano and Dr. Balaji Rengarajan. It has been a great honor to be both of their first PhD student. I appreciate Dr. Domenico Giustiniao's greatly help on advising me on designing, implementing and evaluating the OpenVLC platform and his guides on writing papers, giving presentations, demonstrating demos, and his supports on attending almost all the top conferences in our research area. I appreciate Dr. Balaji Rengarajan on his lending me to do a good research, his guides on thinking good research problem and doing great analysis. Without Dr. Domenico Giustiniano and Dr. Balaji Rengarajan, I will not have the current great achievements during my Ph.D studies.

I would also thank Dr. Joerg Widmer, Dr. Vincenzo Macuso and Dr. Sergey Gorinsky for their supports during the tough three months in 2013 when I had no advisor.

My special thanks goes to my classmates / colleagues Arash Asadi, Allyson Sim, Thomas Nitsche, Adrian Loch, Hany Assasa, Danilo Donno, Guillermo Bielsa, Roberto Calvo Palomino, Aymen Fakhreddine. . . I really enjoy my life in IMDEA Networks with you guys.

Thanks my parents for bringing me to this wonderful world, and my grandparents for raising me up and educating me. Thanks for my younger sister's support when I was a college student.

Lastly, but the most important, I will forever be thankful to my wife. Thank her for taking care of me, which ensures me focus on the research.

# Abstract

Radio-frequency based wireless communications have revolutionized our society. Thanks to the important wireless communication technologies Wi-Fi, LTE, and so on, people can now enjoy high data rate and perversive connection while surfing the Internet. However, new problems and demands are rising in today's wireless networks. Increasing capacity demands are requiring more bandwidth and various wireless radio technologies are exacerbating the spectrum problem. Now technologies and paradigms are needed to meet these needs. In this thesis, I investigate two technologies towards this direction: Visible Light Communication (VLC) and Device-to-Device (D2D) communication.

Although more and more researchers are becoming interested in VLC, the lacking of an open-source platform for VLC research is perverting the fast investigations of VLC. To solve this problem, I design, implement, and evaluate the first open-source platform OpenVLC for embedded VLC research. OpenVLC employs cost-efficient and off-the-shelf optical components and electronics to provide a research platform. The software solutions are developed as a Linux driver and can easily connect to the TCP/IP layers. This allows for the adoption of various Linux diagnostic tools to evaluate the VLC's properties and performance. Based on OpenVLC, I propose a new MAC protocol that enable the intra-frame bidirectional transmissions in networks of visible LEDs. The method adopts only a single LED at each node for both transmission and reception. Through this technology, the system's throughput can be improved a lot and the hidden-node problem can be alleviated greatly. Motivated by the envision of the Internet of lights, I study how to provide stable visible light links in VLC. I identify the limitations and tradeoff of two different types of optical receivers photodiode and LED, and design and implement a new optical data link layer that was resilient to dynamic environments.

On the other hands, to meet the increasing demands, small cells are proposed and deployed in latest cellular networks. As a result, the number of users served by each cell is decreasing. As the opportunistic gain increases as a concave function of active users, in small cells and when dynamic traffic load are considered, the opportunistic gain will lost. To recoup the opportunistic gain, I propose a base-station transparent method based on D2D communication to dispatch traffic among devices. Dynamic programming is used to find the optimal dispatching policy. The results show this method can improve the average packet transfer delay greatly. To increase the opportunistic gain by a further step, I propose a base-station initiated policy to solve the same

problem. An algorithm is therefore designed and implemented, and its performance shows that it can reduce the frame loss ratio significantly.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Wireless communications based on Radio-Frequency (RF) spectrum has revolutionized the way our societies work. Recent deployments of advanced wireless technologies have been able to either provide high data rate (e.g., IEEE 802.11ad), or support high mobility together well high data rate (e.g., LTE), or can operate at a cost of very low energy consumption (e.g., Bluetooth low energy). The so-well understood of the RF communication can also allow to embed tiny wireless transceivers on any *'thing'*, that enable us to build the so-called Internet of Things (IoT).

Nevertheless, new problems and demands in wireless networks are rising. Although nowadays' cellular network LTE can already provide high data rate, new data intensive applications are emerging in the daily routines of mobile users, that make telecom operators struggle to accommodate the increasing demands of mobile traffic. Cellular networks are going to occupy wider bands, exacerbating the *'spectrum crunch'* problem.

Therefore, researchers in wireless communication society are seeking for new paradigms to revolutionize the traditional wireless communication methods. New technologies are under investigated, such as the deep understanding of using 60GHz in wireless communication, new coding/decoding schemes, new type of communication adopting visible light as the transmission medium, device-to-device communication, WiFi-Direct or wireless power communication.

This thesis focuses on two of the new technologies: Visible Light Communication (VLC) and device-to-device communication.

## 1.1. Visible Light Communication System

Wireless communication with visible light is an idea that dates back to the $19^{th}$ century. This old concept is having a strong come back due to two reasons. First, thanks to advancements in VLC, LED lights can now be modulated at high speeds (Kb/s and Mb/s) making them competitive alternatives to wireless communication. Second, due to the high energy efficiency of LEDs, any device that emits light nowadays is likely to be LED-based: car lights, city lights, billboards, toys, wearables and home appliances are already LED-based, just to name a few. Thus, in the

future we could have a new type of pervasive infrastructure waiting to be networked, an Internet of Lights (IoL) [1]. IoL will integrate communication, sensors and light, and create new pervasive smart environments for connected devices and objects, all centered around the light as a medium of communication. Exploiting the visible light spectrum could provide equally disruptive effects. VLC also represents an appealing alternative to RF for networked embedded devices, for instance in the IoT, wearable computing[ref], indoor localization [ref] and vehicular networks [2–4]. In addition, the adoption of VLC would reduce the health hazards caused by overexposure to RF.

VLC experimental research in networked embedded systems (Networked VLC) has yet to gain momentum due to the lack of a low-cost reference platform. The drawbacks of existing experimental work on VLC platforms include its lack of openness, its failure to provide broad support for common networking protocols, and its focus on high-end platforms [5–7]. Besides, the directionality of "optical antenna" of VLC nodes, i.e., the Field Of View (FOV) of LEDs, varies greatly from device to device. An infrastructure, e.g., a light bulb on the ceiling, normally emits light with a wide-FOV. In contrast, mobile devices may have various FOVs, according to the space and power constraints. This variety of light wave propagation calls for networking approaches that are robust to the specific optical antenna. The design opportunities can take advantage of fundamental differences with respect to RF communication.

This thesis takes the first steps to design an open-source, general-purpose platform for embedded research on VLC. Beyond this, this thesis also study the bidirectional transmissions in VLC and analyzes the complexity of visible-light links from a *networking* perspective. It exposes some of the challenges faced by visible light links and propose initial solutions to overcome them.

The structure and contributions of the part "VLC system" enclosed in this thesis are:

- Design, implementation, and evaluation of an open-source platform that can enable VLC research in the filed of networked embedded system [Section II];
- Proposal of a scheme that can achieve intra-frame bi-directional transmissions in a network of LEDs. The scheme can also alleviate the hidden-node problem greatly [Section III];
- Investigation of achieving a reliable link for the Internet of Lights [Section IV].

## 1.2. D2D Communication in Small Cells

To meet the increasing demands on wireless traffic, researchers are seeking for new paradigms to revolutionize the traditional communication methods of cellular networks. D2D communication is one of such paradigms that appears to be a promising component in next generation cellular technologies. D2D communication in cellular networks is traditionally defined as the direct communication between two mobile users without traversing the BS and core network. D2D communication can occur on the cellular spectrum (i.e., *inband*) or unlicensed spectrum (i.e., *outband*).

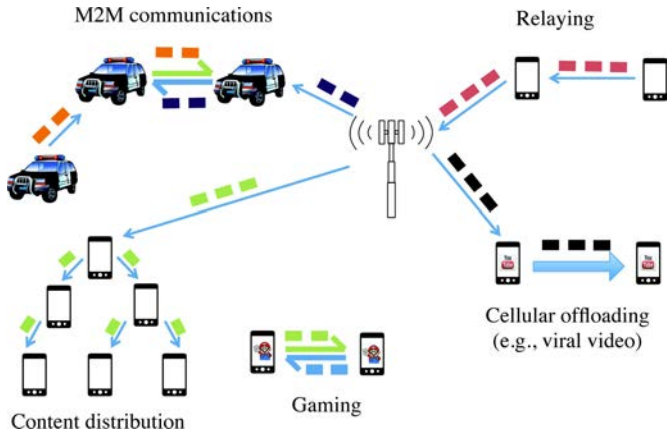



Figure 1.1: Representative use-cases of D2D communications in cellular networks.

In academia, D2D communication was first proposed in [8] to enable multihop relays in cellular networks. Later the works in [9–13] investigated the potential of D2D communications for improving spectral efficiency of cellular networks. Soon after, other potential D2D use-cases were introduced in the literature such as multicasting [14, 15], peer-to-peer communication [16], video dissemination [10, 17–19], machine-to-machine (M2M) communication [20], cellular offloading [21], and so on. The most popular use-cases of D2D communications are shown in Fig. 1.1.

This thesis studies how to use D2D communication to increase the opportunistic gain in small cells. Opportunistic scheduling [22, 23] was proposed for multiuser wireless communication networks to exploit fluctuating channel conditions, aiming to improve performance. In cellular networks, opportunistic schedulers use knowledge of the channels between Base Station (BS) and users to schedule those with favorable channel states, thus improving overall throughput. As cell sizes in future wireless networks shrink in response to increasing demands for capacity [24–26], the average number of users served a BS will decrease. Since opportunistic gain scales as a concave function of the user population [27], presently used scheduling algorithms are prone to losing effectiveness in small cells, especially with dynamic traffic load.

The contributions of this part enclosed in this thesis are as follows:

- Proposal of a BS transparent D2D communication scheme that improve the opportunistic gain in small cells [Section V].

- Proposal of a BS driven D2D communication scheme to improve the opportunistic gain in small cells [Section VI].

## 1.3. Publications Associated with this Thesis

Publications on VLC system:

1. **Q. Wang** and D. Giustiniano. "Intra-Frame Bidirectional Transmission in Networks of Visible LEDs". *IEEE/ACM Transactions on Networking*, accepted on Feb. 12, 2016.

2. **Q. Wang**, D. Giustiniano. "Communication Networks of Visible Light Emitting Diodes with Intra-Frame Bidirectional Transmission". In *ACM CoNEXT*, 2014.

3. **Q. Wang**, D. Giustiniano and D. Puccinelli. "An Open-Source Research Platform for Embedded Visible Light Networking". *IEEE Wireless Communications*, 22(2), 2015.

4. **Q. Wang**, D. Giustiniano, O. Gnawali. "Low-cost, Flexible and Open Platform for Visible Light Communication Networks", In *ACM MobiCom workshop on HotWireless*, 2015.

5. **Q. Wang**, D. Giustiniano and D. Puccinelli. "OpenVLC: Software-Defined Visible Light Embedded Networks". In *ACM Mobicom workshop on Visible Light Communication Systems (VLCS)*, 2014.

6. D. Giustiniano and **Q. Wang**. "OpenVLC, an Open-Source Platform for the Internet of Light". In *IEEE Photonics Society Summer Topicals Meeting Series*, (invited), 2015.

7. **Q. Wang**, D. De Donno and D. Giustiniano. "Demonstration Abstract: Research Platform for Visible Light Communication and Sensing Systems". [Demo] In *ACM/IEEE IPSN*, 2016.

8. **Q. Wang**, S. Yin, O. Gnawali and D. Giustiniano. "Demo: OpenVLC1.0 Platform for Research in Visible Light Communication Networks". [Demo] In *ACM MobiCom*, 2015.

9. **Q. Wang**, D. Giustiniano and D. Puccinelli. "OpenVLC: Software-Defined Open Architecture Embedded Visible Light Networks". [Demo] In *the ACM MobiCom workshop on Visible Light Communication Systems (VLCS)*, 2014.

Publications on D2D communication in small cells:

1. **Q. Wang**, B. Rengarajan and J. Widmer. "Increasing Opportunistic Gain in Small Cells Through Energy-Aware User Cooperation". *IEEE Transactions on Wireless Communications*, 13(11), 2014.

2. A. Asadi, **Q. Wang** and V. Mancuso. "A Survey on Device-to-Device Communication in Cellular Networks". *IEEE Communications Surveys and Tutorials*, 16(4), 2014.

3. **Q. Wang**, B. Rengarajan and J. Widmer. "Increasing Opportunistic Gain in Small Cells Through Base Station-Driven Traffic Spreading". In *IEEE WoWMoM*, 2014.

4. **Q. Wang**, B. Rengarajan. "Recouping Opportunistic Gain in Dense Base Station Layouts Through Energy-Aware User Cooperation". In *IEEE WoWMoM*, 2013.

# Part I : Visible Light Communication System

# Chapter 2

# OpenVLC Research Platform

## 2.1. Introduction

As a spectrum-rich alternative to RF, VLC is attracting the interests of both researchers and engineers. It also represents an appealing alternative to RF for networked embedded devices, for instance in the Internet of Things, wearable computing, indoor localization and vehicular networks [2–4]. However, due to the lack of a low-cost reference platform, the researches on VLC are still lagging a lot. The drawbacks of the existing experimental work on VLC platforms include its lack of openness, its failure to provide broad support for common networking protocols, and its focus on high-end platforms [5–7].

Similarly to how the introduction of the Berkeley motes spearheaded networked embedded systems research a decade ago, it is very likely that a general-purpose, low-cost, open VLC platform would pave the way to novel networking research directions. This chapter takes an initial step toward the adoption of VLC in networked embedded systems and introduces OpenVLC, an open-source software-defined networking platform for fast prototyping. OpenVLC runs on a cost-effective yet powerful embedded board, with a unit cost of approximately sixty dollars. The source codes and instructions of OpenVLC are available at the following URL: `www.openvlc.org`.

This chapter presents the design and evaluation of the open-source OpenVLC research platform. OpenVLC interface an LED-based front-end to an embedded Linux platform and provides a set of software-based primitives, such as signal sampling, symbol detection, coding/decoding, carrier sensing, and communication with the TCP/IP layers of the Linux operating system. This chapter further design and implement a basic Medium Access Protocol (MAC) protocol running in software and illustrate its performance evaluation. The objective of this first step toward Networked VLC is to provide a functional research platform that can be easily extended according to the directions of interest.

In its present form, OpenVLC relies on simple off-the-shelf electronic components and only uses a basic Physical Layer (PHY), which can be scaled to use more advanced PHYs. Cur-

rently, OpenVLC can achieve a MAC layer throughput in the order of the basic rate of IEEE 802.15.7 [28], and UDP throughput of 12.5 kb/s, operating at distances up to 4 m.

The rest of this chapter is organized as follows. Related work are summarized in Sec. 2.2. The system design and implementation of OpenVLC are presented in Sec. 2.3, followed by the evaluation at MAC layer and at system level given in Sec. 2.4. Techniques that could improve the performance of OpenVLC and several research directions that can benefit from OpenVLC are discussed in Sec. 2.5. Closing remarks are presented in Sec. 2.6.

## 2.2. Related Work

Some initial efforts in the embedded systems community have underscored the potential of VLC, such as the investigations on point-to-point communication using smartphones [29**?** , 30], cars [4, 30, 31], and toys [32]. It is also receiving strong attention from the designers of the next generation of cellular networks [33]. VLC's potential usages, challenges of implementation and commercialization, and market conditions are discussed in [34].

The IEEE has developed the 802.15.7 standard [28] for short-range communication using visible light. This standard specifies three PHY layers, which support data rate varying from 11.67 kb/s to 96 Mb/s. It also supports dimming and light flicker prevention. The work in [35] shows an implementation of 802.15.7 protocol using a standard software-defined radio platform from Ettus Research. The implementation cost of the solution is however of at least one magnitude higher than our target platform. In addition, OpenVLC targets the implementation of a platform for networked VLC.

While photodiodes are normally used as receivers, in [36] a reverse-biased LED (rather than a photodiode) is used as a receiver to implement a bidirectional LED-to-LED communication. This principle has been exploited by [37] to introduce a LED-to-LED communication network. The authors study fundamental issues of the design of a low-complexity embedded solution, such as efficient collision detection MAC protocol and light flicker elimination. They show that their prototype can achieve a data rate up to 900 b/s and the communication distance is up to 0.9 m. The work is further developed in [38]. Both [37] and [38] operate on microcontrollers and do not support TCP/IP stack and the wide range of networking protocols available in Linux systems. Their design also has the drawback of being sensitive to noise, since it operates with small unamplified currents.

None of the above works is intended to be open to the research community.

## 2.3. OpenVLC System Design

OpenVLC is a general-purpose software-defined platform for networked VLC. The prototype of OpenVLC is shown in Figure 2.1. It is built around the BeagleBone Black (BBB) board[1], a

[1] `http://beagleboard.org/Products/BeagleBone+Black`

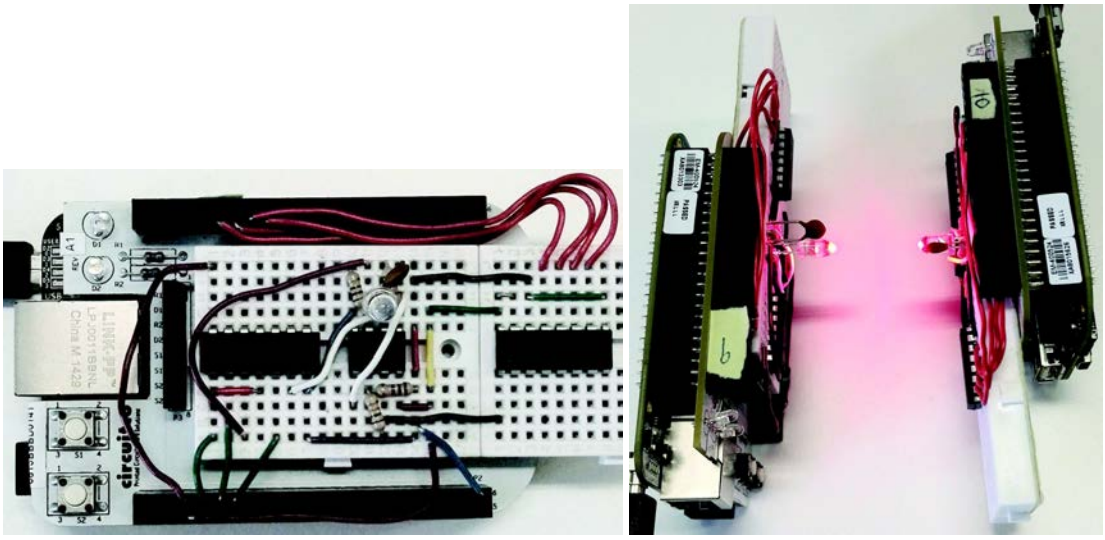


Figure 2.1: The current prototype of OpenVLC: the front-end transceiver of an OpenVLC node is shown on the left, and an example of inter-node communication is shown on the right.

cost-effective, user-friendly, versatile single-board computer with a small form factor. OpenVLC consists of a BBB board, a VLC front-end transceiver and a software-defined system implementation. The front-end transceiver adopts a single LED together with a few basic electronic components for both transmission and reception. OpenVLC's software components are implemented as a Linux driver that communicates directly with the LED front-end and the Linux networking stack. As a result of this design choice, the VLC communication interface can take advantage of the vast range of Linux tools. The communication between two OpenVLC nodes is illustrated in Figure 2.1.

### 2.3.1. Bidirectional Communication

The current version of OpenVLC front-end transceiver reuses the same LED for both transmitting and receiving light signals. Using LEDs as receivers can reduce the design complexity and increases the resilience to ambient noise (e.g., sunlight and indoor illumination [37]) with no need for additional optical filters [39]. The current design can be extended to use photodiodes as receivers, as will be discussed in Sec. 2.5.

The block diagram of the transceiver is shown in the right part of Figure 2.2. It includes a TransConductance Amplifier (TCA) for transmission, a TransImpedance Amplifier (TIA) and an Analog-to-Digital Converter (ADC) for reception, a tristate-output buffer and ancillary circuitry for transmission and reception. A software-defined Transmitter (TX)/Receiver (RX) switch is used to change the LED operation mode between TX and RX through the GPIO pins:

- In **TX mode**, the tristate buffer is enabled and encoded signals are first amplified by the TCA and then fed to the forward-biased LED.

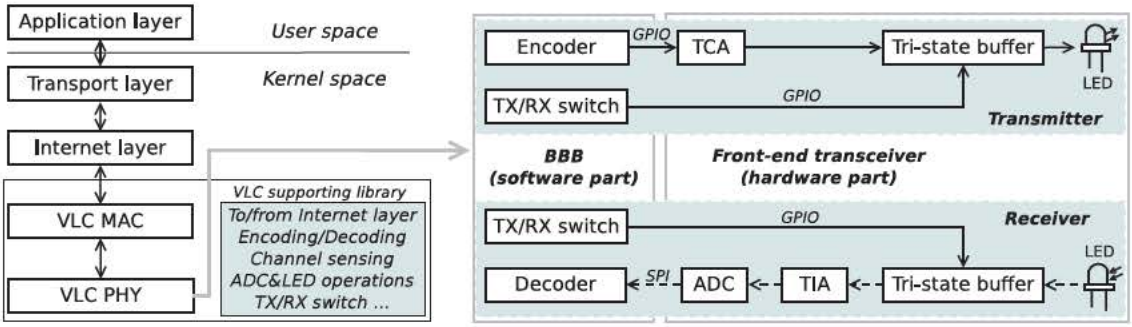



Figure 2.2: Diagram of the front-end transceiver (right) and the communication stack of OpenVLC in an embedded Linux operating system (left).

- In **RX mode**, the tristate buffer is disabled to avoid current leakages to the TCA circuitry, and the light signal is received by the reverse-biased LED. The small photocurrent is then amplified by the TIA. Finally, an ADC converts the output analog signals to digital signals, which are then sent to the decoder through the Serial Peripheral Interface (SPI).

Through the TX/RX switch and the tristate buffer, OpenVLC can switch the LED between being TX mode and RX mode with low latency, such that it can reliably sustain the operation mode for one or more symbol periods. This design offers a basic setup to implement bidirectional communication using a single LED for VLC networks.

### 2.3.2. Software-Defined PHY Layer

The communication stack of OpenVLC is illustrated in the left part of Figure 2.2. Primitives are implemented to build various PHY and MAC layer protocols in the Linux operating system.

**TX, RX and TX/RX switching.** In TX mode, the BBB outputs the signal to the anode of LED for a symbol period. In RX mode, the small photo-current is amplified by the TIA and then sampled by the ADC and converted into a digital signal. The BBB samples the output of ADC at a fixed interval equal to one symbol period. Symbol boundaries are obtained via the real-time timer of the Linux kernel and handled by our driver. When the timer expires, the TX outputs the signal of the symbol waiting to be transmitted and hold the signal for a symbol period. In turn, the RX samples the output of ADC and stores the value in a sequence that will be decoded by the driver at a later time. The LED switches between TX and RX mode through the software-defined TX/RX switch that runs on the BBB.

**Modulation and detection.** OpenVLC adopts intensity modulation for data transmission. Binary information is mapped to the presence (symbol HIGH) or absence (symbol LOW) of the visible light carrier. At the transmitter, OpenVLC uses the On-Off Keying (OOK) modulation and the Manchester Run-Length Limited (RLL) code. Therefore, bit 1 is mapped to symbol sequence LOW–HIGH, and bit 0 is mapped to HIGH–LOW. At the receiver, demodulation is performed with direct detection. Based on the measured voltage, the receiver detects a received signal as a sequence of symbols HIGH and LOW that are then converted to binary data.

**Preamble.** The PHY layer transmits each frame with a fixed-length preamble, consisting of an alternate sequence of HIGH and LOW starting with a HIGH symbol. The numbers of HIGH and LOW symbols in the preamble are the same. To convert symbols into binary data, an adaptive symbol detection threshold is adopted because the received light intensity is greatly affected by the free path loss attenuation of light transmitted from the TX to the RX. This detection threshold is obtained on a per-frame basis by averaging out the digital samples of the preamble sequence. A Special Frame Delimiter (SFD) field is appended to the end of the preamble.

### 2.3.3. Software-Defined MAC Layer

OpenVLC defines two types of MAC frame: DATA and Acknowledgement (ACK). The frame format is shown in Figure 2.3. If the frame has no payload (Length=0), it is inferred to be an ACK. Otherwise, it is a DATA frame. Each frame can carry a payload from 0 to MAX (a predefined value) bytes. The destination and source addresses follow the Length field and each occupies 2 bytes. The 2-byte field Protocol identifies the upper layer protocol encapsulated in the frame payload. Fields from the Length to the Protocol form the MAC header. A two-byte Cyclic Redundancy Check (CRC) over the MAC header and payload is appended after the payload. The Reed-Solomon (RS) error correcting code over the MAC header, payload, and CRC is appended to the end of each frame.

**Carrier sensing.** Wireless MAC protocols usually employ carrier sensing to reduce collisions. In the OpenVLC platform, it provides two types of carrier sensing: *basic sensing* and *fast sensing*. Both are implemented in the PHY layer and can be invoked by the MAC layer. In basic sensing, the platform reads a certain number of continuous symbols. The channel is assessed to be busy if one or more symbols are detected as HIGH symbols; otherwise it is assessed to be clear. Unlike basic sensing, fast sensing operates on per-symbol basis. The channel is assessed to be clear if the symbol is detected as LOW and is assessed to be busy otherwise.

**MAC access protocol.** OpenVLC implements a MAC layer protocol based on the primitives discussed above. It employs a contention-based Carrier Sensing Multiple Access/Collision Detection (CSMA/CD) MAC protocol to ensure fair channel access among all VLC nodes and reduce the impact of collisions [37]. When a frame is ready for transmission, the MAC first calls the *basic sensing* block of the PHY layer. The frame is transmitted immediately if the PHY layer reports the channel is clear. If the channel is assessed to be busy, the MAC starts a backoff counter. The counter is initialized with an integer value randomly drawn from a uniform distribution within the range (0, CW-1]. The contention window CW is initialized as CWmin, where CWmin is the smallest size of the contention window. The PHY layer keeps sensing the channel and each time the channel is assessed to be clear, the counter is decremented. The frame is transmitted when the counter reaches zero.

Upon frame transmission, the transmitter can engage in *fast sensing*. This occurs when the transmitter sends a LOW symbol of the Manchester code, as it powers down the LED and is therefore able to switch the LED to RX mode to receive a symbol, as presented in Sec. 2.3.2.

The received symbol is sufficient for *fast sensing*. Afterwards, the LED is switched back to TX mode to carry on the transmission. The transmitter alternates between TX and RX mode during data transmission. If the transmitter detects a collision, i.e., the channel is assessed to be busy through *fast sensing* for no less than a predefined interval, the ongoing transmission is immediately interrupted. The illustration of the backoff mechanism, basic sensing, and fast sensing in CSMA/CD is shown in Figure 2.3.

After successfully receiving a frame, the receiver sends an ACK to the transmitter. If the transmitter has not received an ACK within the timeout, it retransmits the frame and doubles the CW (until it reaches a pre-defined CWmax threshold that denotes the maximal size of the contention window). The frame is dropped after a pre-defined number of failed retransmissions.

**Interfacing with the Internet layer.** OpenVLC implements the MAC protocol as well as part of the PHY layer as a new driver of the Linux operating system. The MAC protocol will become transparent to various applications if it can connect with the Internet layer. OpenVLC implements two primitive functions to receive a packet from the upper layer and the PHY layer, respectively. The first function is called by the Internet layer to move packets to the MAC layer, where they are enqueued for transmission scheduling. The second one receives packets from the PHY layer, checks their protocols, and decides whether or not to send them to the Internet layer. By invoking these two functions, any MAC protocol can easily interact with the Internet layer.

## 2.4. Experimental Evaluation

The experimental evaluation described in this section uses the Debian Linux Distribution with kernel version 3.8.13 and the Xenomai patch. Details of the electronic devices employed in the current version of OpenVLC are listed in Table 2.1. Unless otherwise specified, each node uses a symbol period of 20 $\mu$s and (216, 200) Reed-Solomon error correction code. All the experiments

| Preamble | SFD | Length | Dst | Src | Protocol | Payload | CRC |
|---|---|---|---|---|---|---|---|
| 3B | 1B | 2B | 2B | 2B | 2B | 0–MAX B | 2B |

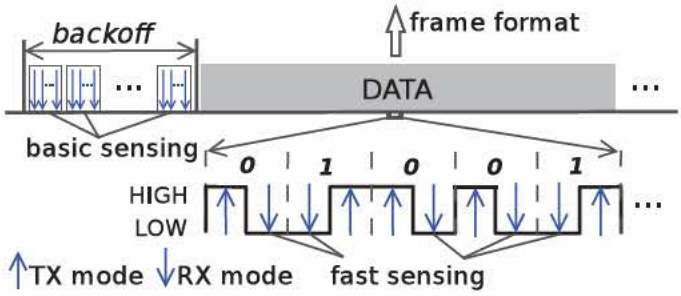



Figure 2.3: Backoff, basic and fast sensing of the CSMA/CD protocol. The transmitter uses OOK with Manchester coding to send data. In the frame format: Length$>$ 0 $\Longleftrightarrow$DATA; Length$=$ 0 $\Longleftrightarrow$ACK.

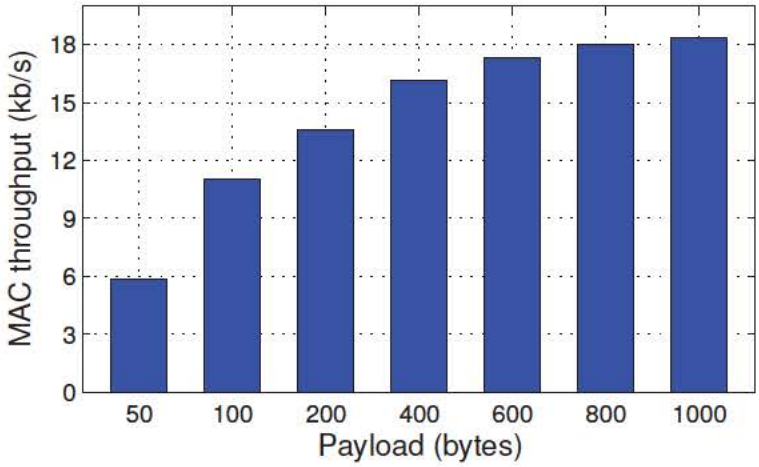



Figure 2.4: MAC layer throughput as a function of the per-frame payload.

are carried out in an indoor office environment in the presence of artificial lighting.

### 2.4.1. MAC layer

This section evaluates the saturation throughput of OpenVLC's MAC layer in a two-node scenario, where the two nodes are within each other's Field-Of-View (FOV) and one continuously transmits to the other. The throughput as a function of the per-frame payload is shown in Figure 4.12, where the two nodes are placed at a distance of 0.6 m and the payload ranges from 50 to 1000 bytes. The saturation throughput is measured up to 18 kb/s. The throughput increases as the payload increases, ranging from 6 kb/s when the payload is 50 bytes to 18 kb/s when the payload is increased to 1000 bytes, which also shows the reliability of software synchronization implementation as frame size gets longer.

### 2.4.2. System Level

It is possible to evaluate the performance of OpenVLC using various traditional network measurement tools. This subsection presents evaluation results obtained from the well-known network

Table 2.1: ELECTRONIC DEVICES USED IN THE PLATFORM.

| Model | Description |
|---|---|
| HLMP-EG08-YZ000 | Low-power 5 nm red LED with a Field-Of-View (FOV) of 8° |
| 74HCT244N | 8-bit buffer with tri-state outputs |
| 2N3904 | Transresistance amplifier |
| LM358N | Transimpedance operational amplifier |
| MCP3008 | 10-bit analog-to-digital converter |

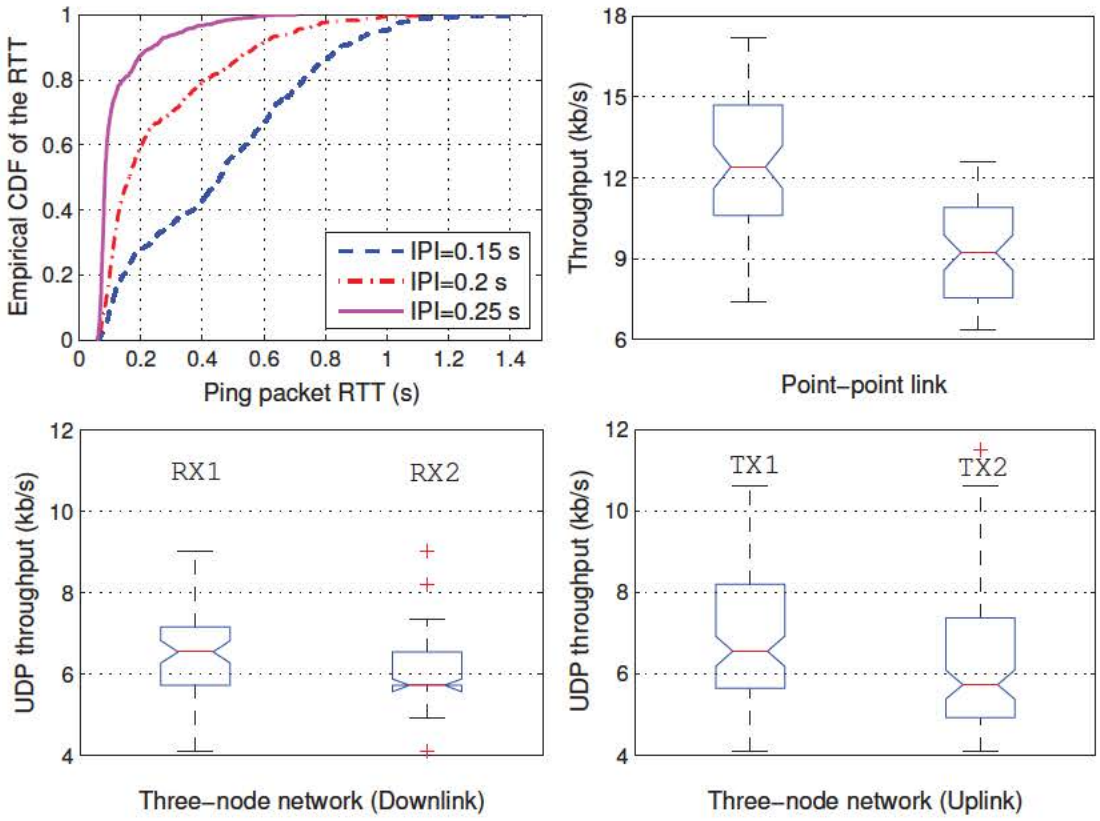



Figure 2.5: System-level evaluation results using `ping` and `iperf`.

tools `ping` and `iperf` in point-to-point link and three-node scenarios.

The OpenVLC's `ping` performance in a point-to-point link scenario over various `ping` Inter-Packet Interval (IPI) settings is shown in Figure 2.5(top-left). These results are collected from 1000 `ping` packets where each `ping` packet has 10-byte data. From the empirical Cumulative Distribution Function (CDF) of the Round-Trip Time (RTT)), it is easy to observe that when the IPI is set to 0.25 s, about 90% of the packets incur a RTT below 200 ms. This value drops to 60% when the `ping` traffic load increases to IPI=0.2 s as a result of the longer queuing time.

The network testing tool `iperf` is also used to evaluate the performance of OpenVLC, and the evaluation results of UDP and TCP over a point-to-point link are shown in Figure 2.5(top-right). Here the UDP datagram size is set to 1000 bytes. The results are collected by running the experiment for 10 minutes in each scenario and the results reported by `iperf` every 10 seconds are plotted. It is observed that the maximal and median achievable throughputs with UDP are about 17 kb/s and 12.5 kb/s, respectively. As for TCP, the maximal and median achievable throughputs are around 13 kb/s and 9 kb/s, respectively. The throughput drops with TCP with respect to UDP is due to the overhead and reliability features of TCP.

Furthermore, the performance of OpenVLC in a three-node (point-to-multipoint) network is evaluated, and the results are shown in Figure 2.5(bottom). In the downlink scenario, one node sends UDP data to the other two nodes (RX1 and RX2). It is observed that the median values of

the UDP throughput of RX1 and RX2 are around 6 kb/s. In the uplink scenario, two nodes (TX1 and TX2) compete for the shared medium to send data to the third one. The median values of the achieved throughput are also around 6 kb/s, which shows a fair access to the medium.

## 2.5. Performance Enhancements and Future Research Directions of OpenVLC

Currently, OpenVLC is designed using basic commercial off-the-shelf components to implement communication network among LEDs. The achieved data rate is already comparable to the lowest one specified in the IEEE 802.15.7 standard, which specifies a PHY rate of at least 11.67 kb/s [28]. In its present form, OpenVLC already offers a flexible starter kit for VLC research.

While most of the VLC efforts so far have targeted point-to-point systems between resource-rich high-end nodes, to date, resource-poor low-end nodes are instead an unexplored research area. Exploring networked systems of resource-poor low-end nodes would be instrumental to the adoption of networked VLC and would require a fundamental redesign of the communication stack. The performance of OpenVLC can be improved to reach out other domains of research investigations, using more powerful hardware and by customizing the software implementation to the application scenarios of choice. This section discusses a set of possible enhancements for OpenVLC as well as future research directions based on it.

### 2.5.1. Performance Enhancements of OpenVLC

This subsection continues to reviews a list of points that could be implemented to boost OpenVLC's performance.

*Matched filtering and timing error recovery* have not yet been implemented in OpenVLC. A matched filter serves to maximize the signal-to-noise ratio and minimize the symbol error probability. Timing error recovery is very useful when the transmitter and the receiver are unsynchronized. To support matched filtering, the front-end transceiver hardware needs to be upgraded. To implement timing error recovery, the software-defined PHY layer of OpenVLC needs to be enhanced to detect the timing error and recover from it. The implementation of the matched filtering as well as the timing error recovery on OpenVLC would also help to increase the communication range as well as the overall system stability for higher rate communication.

The *coverage* of an OpenVLC node is currently limited by the output power and FOV of its LEDs. The output power can be increased by using high brightness white LEDs as optical front-end. For scenarios where one OpenVLC node acts as an access point, hardware should be extended to support Multiple Input Multiple Output (MIMO) LED communication, with modulations such as optical GSSK [40]. This direction exploits the fact that multiple LEDs are usually required for illumination due to the limited brightness of an individual LED. The software would

also need to support the selection and use of different LED-to-LED links.

The current prototype adopts the basic OOK modulation, but *advanced modulation schemes* can also be used by adding a Digital-to-Analog Converter (DAC) or by exploiting the Pulse-Width Modulation (PWM) pins of the BBB. In this way, the disadvantage of OOK in terms of inefficient bandwidth usage can be circumvented. For OpenVLC, the current bottleneck of the achievable date rate is the speed at which the BBB reads symbols from the ADC (the Linux system fails to provide accurate timing past a certain speed [41]). In turn, the BBB can write symbols to the LED at a much faster speed. To eliminate the current bottleneck, *Field-Programmable Gate Arrays (FPGAs)* (as the one used in [42] that can be interfaced with the BBB) or *micro-controllers (MCUs) could be employed for the PHY layer implementation.* Using such solutions, however, would increase the cost of OpenVLC. A cheaper alternative is to use the *Programmable Real-time Unit (PRUs)* of the BBB for dedicated implementation of time-critical functionalities. The ARM CPU of the BBB has two PRUs and each PRU is a low-latency 32-bit micro-controller. To improve the performance of OpenVLC, the PRUs can be used to implement time-sensitive sampling. Because the PRUs can operate at 200 MHz, the performance gain from using them would be significant.

Table 2.2: Summary of possible performance enhancements and research directions of OpenVLC.

| Performance Enhancements & Future research directions | Benefits for the system | Need hardware changes? | Need software changes? | Difficulty |
|---|---|---|---|---|
| Matched-filtering and timing error recovery | -Stability<br>-Throughput<br>-Communication distance | Yes | Yes | Medium |
| Communication coverage (High brightness LED & MIMO) | -Communication coverage | Yes | Yes | Medium |
| Advanced modulation scheme | -Throughput | No/Yes | Yes | Medium |
| FPGA for the PHY | -Throughput | Yes | Yes | Hard |
| MCU for the PHY | -Throughput | Yes | Yes | Medium |
| PRUs for the PHY (kernel space) | -Throughput | No | Yes | Hard |
| PRUs for the PHY (user space) | -Throughput | No | Yes | Medium |
| LED-to-photodiode communication | -To be verified | Yes | Yes | Easy |
| OpenVLC as an app | -Fast testing and deployment | No | Yes | Hard |
| Intra-frame bidirectional transmissions | -Throughput | No | Yes | Medium |
| Integration with RF communication | -Stability<br>-Communication flexibility | Yes | Yes | Hard |

### 2.5.2. Future Research Directions Based on OpenVLC

This subsection discusses a number of promising research directions that can be pursued based on the OpenVLC.

- *LED-to-Photodiode communication:* extending OpenVLC to support LED-to-Photodiode communication is straightforward. It would be very valuable to compare the performance of LED-to-LED and LED-to-Photodiode communications, in terms of transmission range, maximal achievable throughput, resilience ability to ambient light, etc.

- *OpenVLC as an app:* recent research has explored the feasibility of implementing the PHY and MAC layers of ZigBee and WiFi as downloadable pieces of software (such as apps for smartphones) [43]. This approach would streamline the testing and deployment of modifications to existing protocols and, in principle, new protocols as well. With the PRUs of the BBB, it is possible to develop a software on the MAC/PHY protocols of OpenVLC within the user space of Linux without sacrificing the achievable data rate.

- *Enabling intra-frame bidirectional transmissions:* a basic choice for the PHY layer of a VLC system is the OOK modulation with the Manchester Run-Length Limited (RLL) line code. RLL line codes are used to prevent flickering. With the OOK modulation and RLL line codes, a transmitter normally does not need to emit light when it transmits a LOW symbol. As presented in Sec. 2.3.3, the transmitter can then switch the LED to RX mode to receive a symbol. Furthermore, if the receiver has detected a HIGH symbol in current symbol slot and the HIGH symbol is the first part of a modulated bit, then the receiver can switch the LED to TX mode to transmit a symbol during the next symbol slot. Therefore, the receiver can *embed data* into the current frame it is receiving. This technique enables intra-frame bidirectional transmissions that can increase the system throughput to a significant extent. This has been successfully implemented this technique using OpenVLC and the details can be found in Chapter 3.

- *Integration with RF communication:* In order to provide backward compatibility with previous embedded systems, one may think of designing hybrid communication networks that are built on top of both visible light and RF communication. This may allow to exploit the advantage of both technologies, and use them in the most appropriate channel and network conditions.

A summary of these research directions together with the performance enhancements of OpenVLC is given in Table 2.2.

## 2.6. Summary

This chapter presented the design, implementation, and performance evaluation of OpenVLC, an open source platform designed to enable VLC research in the field of networked embedded

systems. OpenVLC's paramount goal is to demystify VLC and lower the barriers to entry to VLC research for embedded systems researchers. Much like the Berkeley motes demystified low-power wireless a decade ago and paved the way to a decade's worth of rich and active research in wireless sensor networks, it is believed that an open reference platform may open up the unexplored area of networked VLC for embedded devices. OpenVLC leverages the recent diffusion of powerful but cost-effective embedded Linux platforms to provide a reference platform that can be used jointly with a vast array of Linux tools. OpenVLC also shows how a handful of commercial off-the-shelf components can suffice as a starter kit for VLC research. Going forward, OpenVLC is expected to serve as a bridge between the VLC community and the wireless embedded systems community. It can be envisioned that research groups in embedded systems with no prior VLC experience can use OpenVLC to explore the realm of visible light, while research groups with a solid VLC background can easily expand OpenVLC and enrich its set of functionalities, for instance with more sophisticated hardware and more advanced PHY designs.

# Chapter 3

# Intra-Frame Bidirectional LED-to-LED Communication

## 3.1. Introduction

VLC is emerging as a complementary technology to mainstream research on RF communication. VLC utilizes visible light from LED to convey digital information between devices. A network of visible LEDs could be enabled by connecting various devices such as ceiling bulbs, lamps, light emitters embedded into cars and mobile devices and perhaps, in the future, LED TVs. However, the directionality of the "optical antenna" of VLC, i.e., the Field Of View (FOV) of LEDs, varies greatly from device to device. An infrastructure, e.g., a light bulb on the ceiling, normally emits light with a wide-FOV. In contrast, mobile devices may have various FOVs, according to the space and power constraints.

This variety of light wave propagation calls for networking approaches that are robust to the specific optical antenna. The design opportunities can take advantage of two fundamental differences with respect to RF communication.

- First, VLC often adopts the On-Off Keying (OOK) modulation or the Variable Pulse Position Modulation (VPPM), thus a transmitter can be "idle" (doest not need to emit light) when transmitting an "OFF" signal. This implies that *other communications could be established during these short times without light emission*. These concurrent communications may improve the system performance, such as increasing the throughput and so on.

- Second, while photodiodes are normally used as receivers, a LED has been proved to work as a receiver in LED-to-LED communications [36, 37]. Thus, a network of LEDs would only require one LED as optical antenna at each transceiver. The challenge is then how to create a network of LEDs *with* different FOVs and *without* additional optical components.

As illustrated in Figure 3.1, these differences $i$) bring opportunities to design new Medium

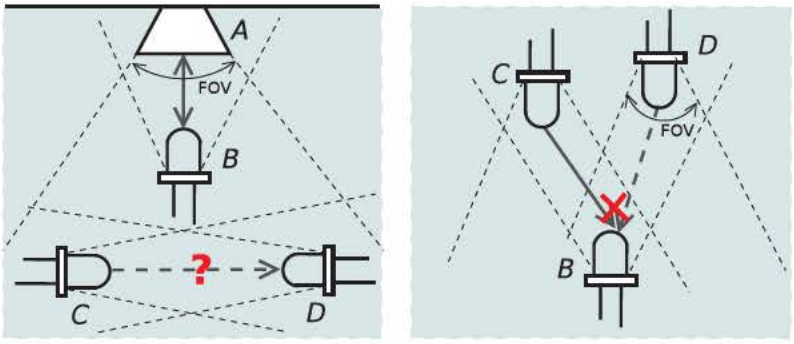



Figure 3.1: Motivation – various "optical antenna" directions of VLC nodes and OOK modulation: *left)* is in-band interference-free concurrent transmissions of $A \leftrightarrow B$ and $C \rightarrow D$ possible? *right)* exacerbate the hidden-node problem.

Access Control (MAC) protocols to improve the network performance; *ii*) increase the hidden-node problem in some scenarios. This chapter proposes a Carrier Sensing Multiple Access/-Collision Detection&Hidden Avoidance (CSMA/CD-HA) MAC protocol to enable intra-frame bidirectional VLC in a network of LEDs. Similarly to the emerging full-duplex of RF communications with a single antenna proposed by [44], CSMA/CD-HA only uses one "antenna" – a single LED to enable in-band bidirectional communication. In contrast to full-duplex research in RF domain, this work targets visible light spectrum and implement very simple techniques to enable intra-frame bidirectional communication with off-the-shelf credit-card-sized embedded board with a total unit cost (including the VLC transceiver) of approximately 50 dollars.

By enabling in-band bidirectional transmissions, the proposed mechanism can improve the system saturation throughput. In ad-hoc networks, the proposed CSMA/CD-HA protocol can shorten the average frame collision time as well as alleviate the hidden-node problem without using the Request-To-Send/Clear-To-Send (RTS/CTS) method, thus improving the system performance in terms of frame collision probability and throughput. This work evaluates the performance of our protocol from analysis and experiments. For the experimental evaluation, this work implements the CSMA/CD-HA protocols and underlying techniques in a software-defined embedded platform running Linux. Experimental results show that the proposed protocol can increase the saturation throughput by nearly up to 50% and 100% in a two- and four-node networks, respectively. In ad-hoc networks, shortened frame collision time is demonstrated by the experimental evaluation. The hidden-node problem is also alleviated greatly, and the system throughput is improved significantly. Practical issues for the implementation, such as the robustness to the data pattern of the payload, are also addressed and solved.

The rest of this chapter is organized as follows. Related work is summarized in Sec. 3.2. Fundamental designs of the intra-frame bidirectional transmission and the proposed CSMA/CD-HA protocol are presented in Sec. 3.3, followed by the analysis and numerical results in Sec. 3.4. Details of the implementation in an embedded platform and the performance evaluations are given in Sec. 3.5 and Sec. 3.6, respectively. Conclusions are drawn in Sec. 3.7.

## 3.2. Related Work

VLC has received strong attention from designers of next generation cellular networks [33, 45], the point-to-point communication using smartphone [29, 46] and cars [4, 31]. Using visible light to enable indoor localization [3, 47, 48], light-to-camera [3, 29, 46, 47, 49] and screen-to-camera [46, 50, 51] communications are also well investigated. Besides, the IEEE has developed the 802.15.7 standard [28] for short-range communication with visible light.

**Single antenna VLC.** While photodiodes are normally used as receivers, a reverse-biased LED instead of a photodiode was used in [36] as a receiver to implement a bidirectional communication network. This principle had been exploited by [37] to introduce a low-power LED-to-LED communication network. This work operated on microcontrollers and was implemented as embedded software in the environment of non-operating system. Connecting it with various networking protocols is not straightforward. In contrast,last chapter has designed and implemented OpenVLC [52], an open-source software-defined platform for VLC networks. OpenVLC is built around a low-cost embedded Linux platform. Its software-defined solution is implemented as a Linux driver and thus it appears as a normal network interface that could easily interoperate with Internet protocols. Recently, authors in [39] investigated the feasibility of adopting a commercial high-power LED as a transceiver. They demonstrated a system that achieves a PHY layer rate of 15 Mb/s with the OOK modulation.

**Full-duplex RF/VLC communication.** In-band full-duplex RF communication was proposed in [53] and further implemented as a prototype in [54]. The key technique is a device that uses the inverse of its transmitted signals to cancel the self-interference to its received signal. Compared to our proposed technique, both of them can achieve in-band bidirectional transmission. A different is that [53, 54] used two antennas for both transmitting and receiving, while in our technique, a single LED is used for that purpose. Recently, a method combining analog and digital cancellation is proposed in [44]. This method enables the full duplex communication with a single antenna. For full-duplex VLC, some authors propose to use visible and infrared lights or LEDs operating on different wavelengths for full-duplex transmission, as in [55]. Furthermore, the authors in [56] propose to use an isolator between the LED and photodiode at each node for full-duplex VLC, and they claim the mutual interference between the bidirectional LED-to-photodiode links is negligible. Compared to these work, here only a single LED is needed at each node to implement an in-band interference-free full-duplex VLC system.

## 3.3. System Design

This sections first briefly introduces some background information on coding and decoding schemes adopted in this work. Then it presents the key technique at symbol level to enable intra-frame bidirectional transmission in a network of visible LEDs, where a single LED is used as optical antenna at each transceiver (without additional optical components). Finally this section

proposes a MAC protocol to exploit this technique at system level.

### 3.3.1. Coding/Decoding schemes

This work uses intensity modulation for data transmission, which is also adopted by the IEEE 802.15.7 standard developed for short-range communication using visible light of wideband light bulbs [28]. Binary information is mapped to the presence (symbol HIGH) or absence (symbol LOW) of the visible light carrier. The main physical layer of the IEEE 802.15.7 standard uses the OOK modulation with the Manchester Run-Length Limited (RLL) line code. RLL line codes are used to avoid long runs of light on and light off that could end up in flicker effects, as well as clock and data recovery detection problems. Therefore, in this system bit 1 is mapped to symbol sequence LOW-HIGH, and bit 0 is mapped to HIGH-LOW. Demodulation is performed with direct detection. Based on the received signal's voltage, the receiving node detects the received signal as the sequence of symbols HIGH and LOW that are then converted to binary data.

### 3.3.2. Key technique at symbol level

The key enabler of the intra-frame bidirectional transmission technique is that a node normally does not need to emit light when transmitting a symbol LOW. Thanks to this, nodes could switch between being a transmitter and a receiver with symbol-level granularity during a frame transmission.

Let us consider an example of the communication between two nodes, A and B. Assume that node A transmits data to node B. *Node A is "idle" when it transmits a symbol LOW, and it can make use of this time to receive a data symbol.* To cooperate with this, node B can start itself to transmit a symbol if it can *predict* that it will receive a symbol LOW in the next symbol slot. An example is Illustrated in Figure 3.2. For the Manchester RLL code used in this paper, the prediction is based on:

- node B receives a symbol HIGH in the current symbol slot;
- the symbol HIGH is the first part of a modulated bit (i.e., bit "0").

Just equipped with one LED, node B can switch to send data during the reception of a frame, while node A can switch to receive data when it transmits a frame. This switching between a transmitter and a receiver at symbol level allows for in-band bidirectional symbol transmission.

As shown in Figure 3.2, node A may expect to receive a symbol only when it transmits the symbol LOW of bit "0". The reason is that node B can only predict the symbol LOW of bit "0", but it can not predict the symbol LOW of bit "1". In expectation, half of data from node A are with bit "1" and the other half with bit "0". Therefore, node B will transmit data for half of the payload of node A.

This approach can be extended to other RLL line codes. In more general terms: the intra-frame bidirectional transmission can be enabled for those symbols that there is a probability of

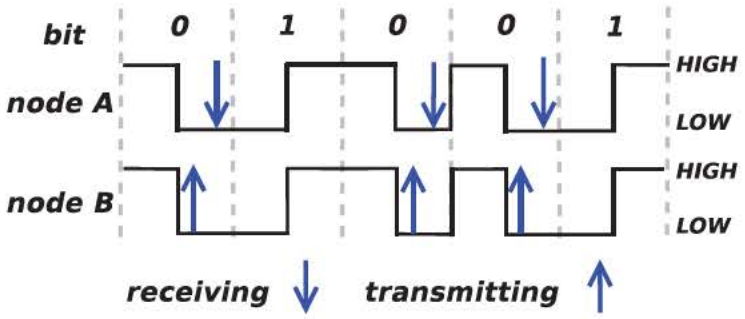



Figure 3.2: The key technique at symbol level that enables intra-frame bidirectional transmission.

one to jump to a state where node A transmits a symbol LOW.

### 3.3.3. The CSMA/CD-HA MAC protocol

The Carrier Sensing Multiple Access/Collision Detection& Hidden Avoidance (CSMA/CD-HA) protocol is proposed to ensure fair channel access among all VLC nodes and reduce the impact of collisions and hidden nodes. When introducing our protocol, this chapter refers to Figure 3.3 for the illustration of the protocol and to Table 3.1 for the short notations.

When a frame is available for transmission, the MAC first senses the channel. The frame is transmitted immediately if the channel is sensed clear. If the channel is assessed busy, the MAC starts a backoff counter. The frame is transmitted when the counter reaches zero. The transmitting node and its corresponding receiver become a primary transmitter (P-TX) and a primary receiver (P-RX), respectively. Based on the symbol-level technique introduced in Sec. **??**:

- The P-TX switches to receiving mode during the transmission of the primary frame when it transmits the symbols LOW of bit "0", waiting to receive the symbols of an embedded frame.

Table 3.1: Notations used in the proposed MAC protocol.

| Notation | Explanation |
|---|---|
| Primary TX (P-TX) | Node that has gained access to the medium by backoff protocol |
| Primary RX (P-RX) | Peer RX of a P-TX |
| Secondary TX (S-TX) | Node (not P-RX) sending an in-band frame with the P-TX's transmission |
| Secondary RX (S-RX) | Peer RX of a S-TX |
| Primary frames | Frames sent by a P-TX to a P-RX |
| Embedded frames | Frames sent opportunistically by either a P-RX or a S-TX |
| Embedded channels | The in-band communication channels where embedded frames are transmitted |

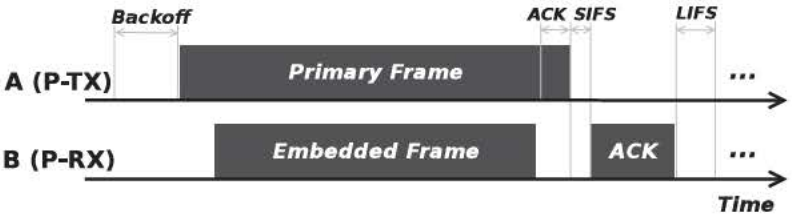



Figure 3.3: CSMA/CD-HA MAC protocol with a primary transmitter and a primary receiver. *(Note: the DATA/ACK frames shown here are transmitted by their corresponding nodes, not received.)*

- The P-RX prepares an embedded frame if it decodes that it is the intended receiver of the primary frame. Every time the P-RX receives a symbol HIGH of bit "0" (thus predicting that the next incoming symbol is LOW), it switches to transmission mode and it sends a symbol of the embedded frame through the embedded channel.

In this way, the intra-frame bidirectional transmission is enabled. Note that the P-RX should transmit the whole embedded frame before the P-TX finishes to transmit the primary frame.

After the P-TX successfully receives the embedded frame, it appends an-octet ACK to the primary frame it is transmitting to acknowledge the reception of embedded data. If the P-RX decodes the frame successfully, it sends an ACK to the P-TX after a duration of Short Inter-Frame Space (SIFS) since it receives the primary frame. After the transmission of the ACK finishes, nodes restart to compete the channel access after a Long Inter-Frame Space (LIFS) period. If the P-TX has not received an ACK within the timeout, a retransmission occurs and the contention window is doubled unless it reaches the maximal contention window. A frame is dropped when a pre-defined number of retransmissions fail.

### 3.3.4. Main features of the CSMA/CD-HA protocol

This part presents the main features of the CSMA/CD-HA protocol. First, *the CSMA/CD-HA protocol has the ability to alleviate hidden nodes.* As a result of intra-frame bidirectional transmission, the P-RX will transmit embedded symbols when it is the intended receiver of a communication initiated by the P-TX. Part of these symbols are HIGH. These symbols HIGH (corresponding to transmission of light) can be sensed by the nodes under its coverage, thus they will not send frames to the receiver during this period and the potential hidden-node problem is alleviated. The only part of the frame which is not protected to hidden nodes is before the embedded frames start to be transmitted, when a hidden node to the TX may judge the channel as clear. However, this duration is usually short compared to the whole frame transmission period and the influence tends to be very low. Sometimes the P-RX may have no data to transmit. Under this case, it can send dummy symbols to announce that the channel is busy while it is receiving a frame. To summarize, the embedded frames of the primary communication in the CSMA/CD-HA protocol have twofold use: send additional in-band data and act as active acknowledgement

**Algorithm 1** Collision detection at the P-RX

**Input:** `txSymb`: to-be-transmitted symbol; `numErr`: the current amount of received invalid sequences; `maxNumErr`: the threshold of `numError`.
**Output:** collision detected: **true / false**; `rxSymb`: the received symbol; `numError`: updated.

1: **if** `txSymb` is HIGH **then**
2: transmit symbol `txSymb` to the P-TX
3: `rxSymb` ← LOW
4: **else**
5: `rxSymb` ← receive a symbol from the P-TX
6: {If P-RX receives an invalid HIGH-HIGH sequence}
7: **if** `rxSymb` is HIGH && (++`numErr`) ¿ `maxNumErr` **then**
8: **return true** {collision detected}
9: **end if**
10: **end if**
11: **return false** {no collision}

**Algorithm 2** Collision detection at the P-TX

**Input:** `numLOW`: the current amount of continuously received symbols LOW; `maxNumLOW`:[2] the threshold of `numLOW`.
**Output:** collision detected: **true / false**; `rxSymb`: the received symbol; `numLOW`: updated.

1: `rxSymb` ← receive a symbol from the P-RX
2: **if** `rxSymb` is LOW **then**
3: **if** (++`numLOW`) ¿ `maxNumLOW` **then**
4: **return true** {collision detected}
5: **end if**
6: **else**
7: `numLOW` ← 0
8: **end if**
9: **return false** {no collision}

of ongoing primary data reception to protect the primary transmitter from hidden nodes.

*CSMA/CD-HA is also useful for collision detection.* Traditional collisions caused by nodes reaching the backoff counter equal to zero are still possible. Collisions may also occur when hidden nodes start transmitting before the P-RX's transmission of an embedded frame, as described in the previous paragraph. *These collisions are detectable if the P-RX receives the HIGH-HIGH sequence (which is invalid) for a pre-defined certain times.* Details of the collision detection at P-RX is presented in Algorithm 1.[1] Upon a collision detected by P-RX (lines 7-9 of Algorithm 1), P-RX stops transmitting embedded data if the transmission of the embedded frame was ongoing, or otherwise it does not start the transmission of an embedded frame. Consequently, *this collision can be detected by P-TX due to the absence of embedded transmission from P-RX*. The detection

[2]The `maxNumLOW` is initiated based on experience, and then is updated on per-frame basis to be twice of the maximal continuous "0" in the latest received frame.

[1]Note that the algorithm is only called by the P-RX when $i$) it receives a symbol HIGH in the previous symbol slot and $ii$) it infers that the symbol HIGH is the first part of a modulated bit.

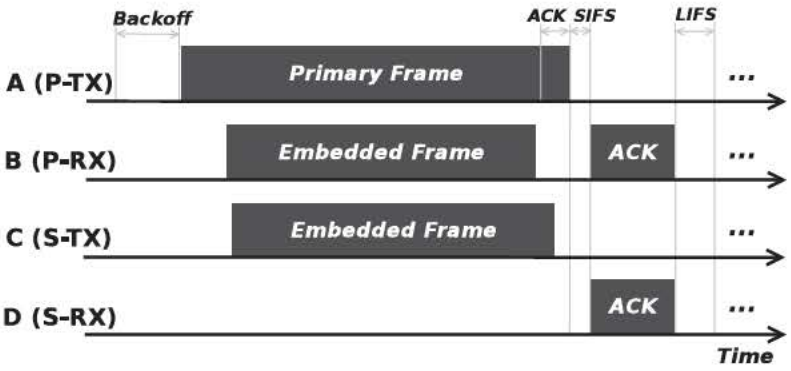



Figure 3.4: CSMA/CD-HA MAC protocol that supports intra-frame bidirectional transmission: an illustration of concurrent transmissions for the left scenario of Figure 3.1. *(Note: the DATA/ACK frames shown here are transmitted by their corresponding nodes, not received.)*

details are shown in Algorithm 2, which is executed by P-TX after it transmits the symbol HIGH of a modulated bit "0" (no need to emit light to transmit the following symbol LOW). If a collision is detected (lines 3-5 of Algorithm 2), the ongoing primary frame transmission will be terminated by the P-TX immediately without waiting for the end of the payload, which increases the channel utilization.

Finally, *secondary intra-frame concurrent transmission can also be enabled,* depending on the number of existing nodes, the FOVs of their LEDs and their positions. Take the left scenario in Figure 3.1 as an example. The wide-FOV LEDs of node A make it possible to communicate with nodes B, C, and D, while the latter three have limited connectivity due to their narrow-FOV LEDs. Let us illustrate an example of the protocol in Figure 3.4. Nodes C and D can receive A's signals, and C can detect that it is not (as well as D) the intended receiver of data from A. Node C then transmits embedded frames to D as node B transmits to A (C thus becomes a S-TX), while D can receive these frames following exactly A's receiving steps (D thus becomes a S-RX). After successfully decoding a frame from node C, D sends back an ACK when the transmission of the primary frame is finished. The transmission of ACK from node D to C will not cause any interference to the communication between nodes A (the P-TX) and B (the P-RX).

### 3.3.5. Discussion

Let us continue to discuss some other important aspects of the proposed protocol.

*Length of an embedded frame:* the length of an embedded frame heavily depends on the number of bit "0"s in the primary frame body. This number can be calculated by the transmitter P-TX based on its payload and appended to the MAC frame header. Note that this appended number only serves as a reference. The actual length of an embedded frame also depends on when the P-RX/S-TX starts transmitting an embedded frame. The P-RX/S-TX calculates the length of embedded frame by subtracting $L$ from the reference number suggested by P-TX, where $L$ is the amount of information P-RX/S-TX can transmit during the time interval between the

transmissions of primary and embedded frames by the P-TX and P-RX/S-TX, respectively. The $L$ increases if P-RX/S-TX spends longer time for the backoff process on competing the secondary channel.

*When to send an embedded frame:* an embedded frame should be transmitted after a node receives the source address of the current primary frame. Only the intended receiver of the transmission can be declared as P-RX. Other nodes may act as S-TX, as long as they are not the intended receivers of P-RX. They start a secondary backoff process that follows the same rules as the primary one to decide the access to the medium.

*Robustness of intra-frame concurrent transmissions:* the intra-frame current transmissions should be robust to expected or unexpected events, i.e., the secondary transmission should be disabled immediately if the primary transmission stops due to detections of collisions or hidden nodes or other unexpected errors. The CSMA/CD-HA protocol has this ability. For example, an underlying assumption of secondary transmissions is that the FOVs of nodes is such that the transmission of S-TX and S-RX does not interfere with the transmission of P-TX and P-RX. If this is not the case, the primary transmission is stopped due to detected collisions. As soon as P-TX stops its transmission because of a detected collision, the secondary transmission of S-TX and S-RX will be disabled for a while.

*Intra-frame bidirectional transmission between S-TX and S-RX*: In our current protocol design, S-RX does not predict "LOW" symbols transmitted by S-TX, to reduce complexity. Therefore, the transmission between S-TX and S-RX is not intra-frame bidirectional, and S-TX could not detect collisions during its transmission of an embedded frame. In principle, however, collision detection by S-TX and the intra-frame bidirectional transmission between S-TX and S-RX are feasible assuming that S-TX adopts a similar modulation scheme (e.g., OOK + Manchester RLL code) as P-TX does.

## 3.4. Throughput Analysis

This section analyzes the system performance of the CSMA/CD-HA protocol in terms of saturation throughput. It first considers a simplified CSMA/CD-HA protocol to analyze the performance improvement from embedded channels by enabling the intra-frame bidirectional transmission. After that, this section analyzes the performance of the CSMA/CD-HA protocol in common ad-hoc scenarios.

### 3.4.1. Performance with embedded channels

To analyze the performance improvement from embedded channels, let us first adopt *a simplified CSMA/CD-HA protocol where the backoff mechanism is disabled.* Consider a network consisting of an access point (AP) equipped with wide FOV LEDs and $N$ users with narrow FOV LED, similar to the scenario presented in Figure 3.1(left). All the users are under the AP's coverage and therefore can detect the data it transmits in the same frequency band of the visible light

spectrum. During the AP's transmission, the users can opportunistically send data to the AP or to other users through the embedded channels as presented in Sec. 3.3. To simplify the derivation, *let us assume that the AP always has access to the channel and it keeps sending data to users, and other users can only transmit data through embedded channels.*

To derive the saturation throughput of this system, let us first make some notations. Let $T_h$ denote the time to transmit the frame header, and $E[T_p]$ be the average time for the transmission of frame payload. The intervals of SIFS and LIFS are denoted as $T_s$ and $T_l$, respectively. Moreover, let $T_a$ be the time to transmit an ACK and $\delta$ be the propagation delay. Assume there is no error in the PHY layer transmission. Then the total time $T_f$ to transmit a frame can be written as

$$T_f = T_h + E[T_p] + \delta + T_s + T_a + \delta + T_l \tag{3.1}$$

Let us assume the AP serves the users alternately. Let $E[L_p]$ denote the average effective payload of a primary frame, and $S_{wo}$ be the system saturation throughput without embedded channels. Then $S_{wo}$ can be written as

$$S_{wo} = N \cdot \frac{E[L_p]}{N \cdot T_f} = \frac{E[L_p]}{T_f} \tag{3.2}$$

Similarly, let $E[L_{em}]$ be the average effective payload of embedded frames, and $S_w$ denote the saturation throughput of a system with an average number of $N_{em}$ embedded channels. Then

$$S_w = \frac{E[L_p] + N_{em} \cdot E[L_{em}]}{T_f} \tag{3.3}$$

### 3.4.2. Ad-hoc scenario

This work adopts the Markov chain model to analyze the performance of the CSMA/CD-HA protocol in ad-hoc networks. It considers a system consisting of $N$ users and all the users can communicate with each other, *i.e.*, there are no *hidden nodes*. This work assumes all nodes always have data to transmit and aims to derive the MAC layer saturation throughput of the system.

The classic Markov Chain model to analyze the performance of the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) MAC protocol of IEEE 802.11 was introduced in [57], and then extended to many other different scenarios, *e.g.*, unsaturated traffic [58], hidden nodes scenario [59], and so on. The author in [57] adopted a two-dimension Markov chain to describe the 802.11 MAC operations.

This model is also used in the system presented in this work, but set a maximal retransmission times and assume that it is equal to the maximal backoff stage. A frame will be dropped if its retransmission times exceed the retransmission threshold. Following the derivation presented in the appendix, it is not difficult to compute the saturation throughput of CSMA/CA, CSMA/CD (for one single optical antenna, and presented in [37]), and the CSMA/CD-HA (with embedded transmission) proposed in this work. Let $S_{ca}$ denote the saturation throughput under CSMA/CA.

Table 3.2: Parameter settings in the analysis.

| Parameter | Value (unit) |
|---|---|
| $N$ | 2, 3, ..., 10 |
| $m$ | 4 |
| $T_h$ | 168 (optical clocks) |
| $T_s,\ T_l,\ T_a$ | 40, 120, 200 (optical clocks) |
| $\sigma$ | 0.001 (optical clocks) |
| $L_p$ | 50, 100, 200, 400, 600, 800, 1000 (bytes) |
| $L_{em}$ | 1, 20, 80, 180, 280, 380, 480 (bytes) |

Then $S_{ca}$ can be written as

$$S_{ca} = \frac{P_f E[L_p]}{(1-P_{tr})\sigma + P_f T_f + P_c T_c^{ca}} \tag{3.4}$$

where $\sigma$ is the duration of an empty time slot. Similarly, let $S_{cd}$ and $S_{cd\_ha}$ be the saturation throughputs under CSMA/CD and under CSMD/CD-HA (with embedded transmission), respectively. Then

$$S_{cd} = \frac{P_f E[L_p]}{(1-P_{tr})\sigma + P_f T_f + P_c T_c^{cd}} \tag{3.5}$$

$$S_{cd\_ha} = \frac{P_f (E[L_p] + E[L_{em}])}{(1-P_{tr})\sigma + P_f T_f + P_c T_c^{cd}} \tag{3.6}$$

### 3.4.3. Numerical Results

This part presents some numerical results of our proposed protocols. The parameter settings used in the calculation are listed in Table 3.2. The payloads of embedded frames $L_{em}$ are set to the maximal lengths that can ensure the transmissions of embedded frames finish before those of primary frames. In addition, the Reed-Solomon (RS) error correcting code (232,200) is applied to all the primary and embedded frames.

#### 3.4.3.1. Performance with embedded channels

Without loss of generality, here considering the scenario illustrated in Figure 3.1(left). This work compares the system saturation throughput under three different cases: $i)$ without intra-frame bidirectional transmission (*i.e.*, no embedded channel); $ii)$ with the intra-frame bidirectional transmission between nodes A and B (*i.e.*, one embedded channel, from node B to A); $iii)$ with intra-frame bidirectional transmissions between nodes A and B, and the intra-frame unidirectional transmission from node C to D (i.e, two embedded channels, from node B to A, and from node C to D.)

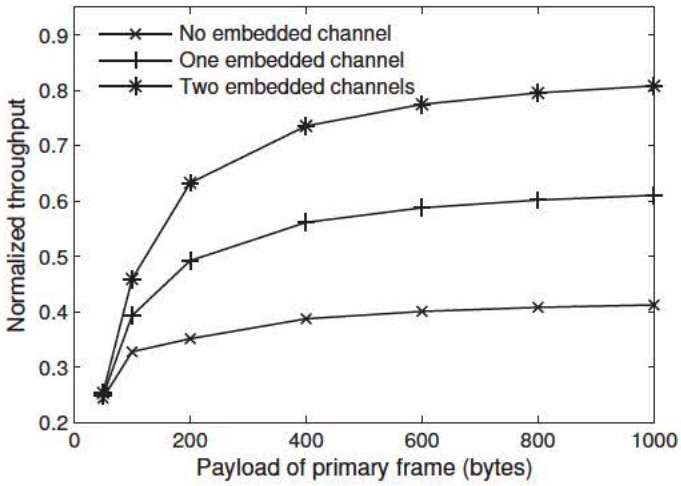


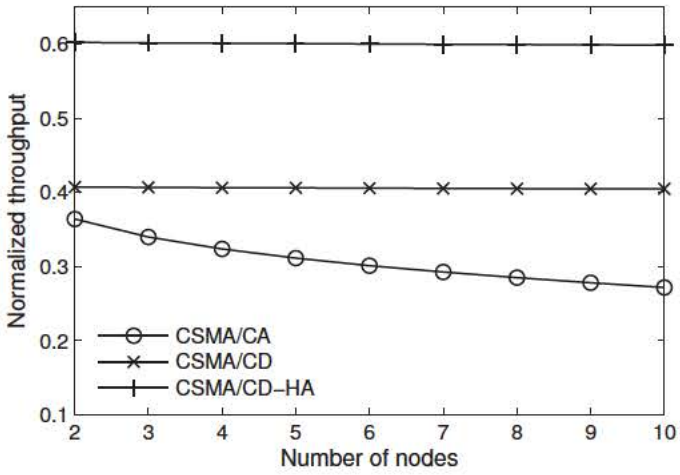

Figure 3.5: Theoretical saturation throughput with embedded channels.

The normalized saturation throughputs versus the payload of primary frames are shown in Figure 3.5. First it can be observed that regardless of the number of embedded channels, the throughput increases consistently with the payload of primary frames, as expected. When the payload of primary frames is small (*e.g.*, 50 bytes), the embedded frames can hardly carry any payload. Thus the throughputs under all cases are similar to each other. It is also observed that the throughput gains from embedded channels increase as the payload of primary frames increases. When the payload increases to 1000 bytes, the throughputs under one and two embedded channels can outperform that under no embedded channel by nearly 50% and 100%, respectively. Besides, it can be concluded from the figure that the throughput gain will increase greatly if there are more embedded channels enabled.

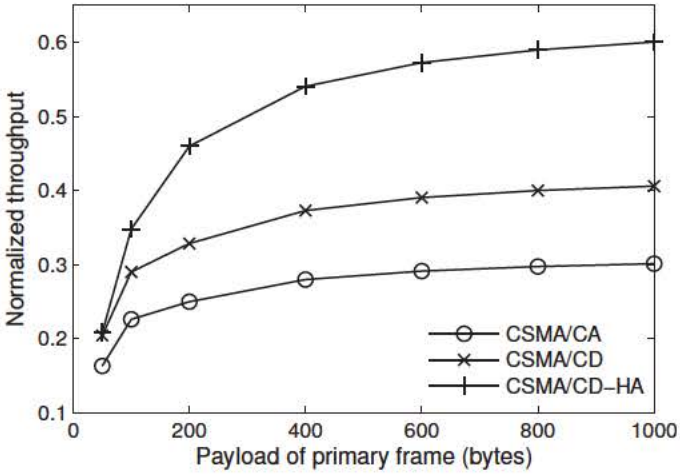
#### 3.4.3.2. Ad-hoc scenario

This subsection compares numerical results in ad-hoc networks under three different MAC protocols: $i$) the CSMA/CA protocol as described in [57]; $ii$) the CSMA/CD protocol [37]; $iii$) the proposed CSMA/CD-HA protocol with enabled intra-frame bidirectional transmission in this work.

Figure 3.6(a) shows the normalized saturation throughput versus the number of nodes. Those nodes compete with each other to access the shared channel. The payloads of each primary and embedded frames are fixed to 1000 and 480 bytes, respectively. First it can be observed that the throughput with CSMA/CA decreases gradually as the number of nodes increases, in accordance with the result shown in [57]. The reason behind those results is that the more nodes compete for the shared channel, the more time will be wasted on transmitting corrupted frames due to collisions. Instead, the throughputs with both CSMA/CD and CSMA/CD-HA are less sensitive to the number of competing nodes. This benefits from the fact that, with CSMA/CD and CSMA/CD-HA, the collisions can be timely detected and the transmissions of corrupted




(a) Throughput vs. number of nodes




(b) Throughput vs. primary frame payload

Figure 3.6: Theoretical saturation throughput in ad-hoc networks.

frames are stopped immediately, saving time from transmitting frames that would waste the shared resources. Thanks to the embedded channel, it can be seen from the figure that CSMA/CD-HA outperforms CSMA/CD by around 50 %.

The normalized saturation throughput versus the payload of primary frames is shown in Figure 3.6(b). The number of nodes is fixed to six. As expected, the throughputs with all protocols increase consistently with the payload. It can also be observed that the throughputs of both CSMA/CD and CSMA/CD-HA are higher than that with CSMA/CA. Besides, the difference between CSMA/CD and CSMA/CD-HA increases as the payload increases, due to the longer payloads carried by embedded frames.

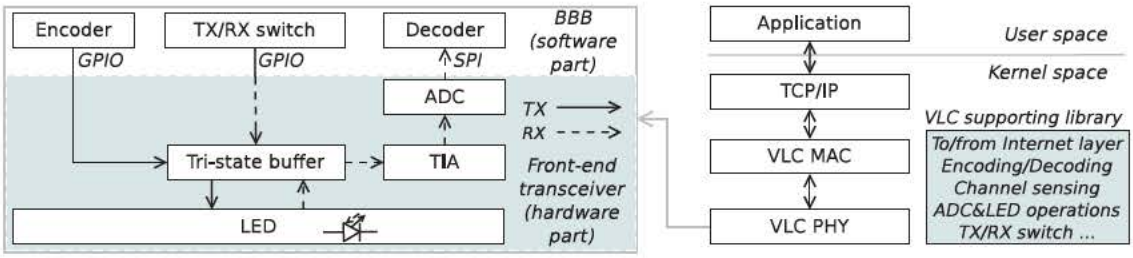



Figure 3.7: The diagram of our implementation: left) the front-end transceiver; right) the communication stack.

## 3.5. Implementation

This work implements the proposed MAC protocol and the underlying symbol-level techniques in the general-purpose software-defined open source platform OpenVLC for visible light communication networks [52]. The implementation is part of a new Linux driver, that is, another network interface, that can communicate directly with the VLC hardware and the Linux networking stack.

### 3.5.1. Hardware

The OpenVLC platform consists of a BeagleBone Black (BBB) board [60] and a front-end transceiver. The transceiver adopts a single LED to both transmit and receive along with a few basic electronic components. This design has been proved to be resilient to ambient light interference without additional electronic processing [37], and could be well suited for a communication network of consumer devices with LED front-end. This work adopts off-the-shelf electronic components to implement a basic physical layer. The block diagram of transceiver is shown in Figure 3.7(left). It includes a TransImpedance Amplifier (TIA) and an Analog-to-Digital Converter (ADC) for reception, a tri-state-output buffer and ancillary circuitry for switching between transmission and reception.

A prerequisite to enable the intra-frame bidirectional transmission is that the nodes can swiftly switch between TX mode and RX mode on a symbol basis. In our implementation, a software-defined TX/RX switch is used to switch the LED between TX and RX through the GPIO pins:

- in **TX mode**, encoded signals are amplified by the tri-state buffer and then fed to the forward-biased LED for light emission;

- in **RX mode**, light signal is received by the LED and then the received small photocurrent is amplified by the TIA.

Finally, an ADC converts the output analog signals to digital signals, which are then sent to the decoder through the Serial Peripheral Interface (SPI).

| Preamble | SFD | Length | Dst | Src | Protocol | Payload | CRC |
|---|---|---|---|---|---|---|---|
| 3 Bytes | 1 B | 2 B | 2 B | 2 B | 2 B | 0-MAX B | 2 B |

Figure 3.8: Frame format of DATA and ACK: Length$> 0 \Longleftrightarrow$DATA; Length$= 0 \Longleftrightarrow$ACK.

Through the TX/RX switch and the tri-state buffer, the LED can switch between being TX and RX modes with low latency. This makes it possible to enable the intra-frame bidirectional transmission presented in Sec. 3.3.2.

### 3.5.2. Software

The operating system running within the BBB board is the Debian Linux Distribution. Figure 3.7(right) illustrates the software stack of the implementation, where the VLC MAC and VLC PHY are built based on the primitive functions implemented. These functions include writing a symbol to the LED, reading a symbol from the ADC, coding/decoding, preamble detection, TX/RX switching, and so on.

In our implementation, there are two types of frames in the MAC layer, DATA and Acknowledgement (ACK). The DATA frame structure and octets each field occupies are shown in Figure 3.8. Distinguishing DATA and ACK frames is through the length of frame body (payload): if the frame has no payload (i.e., "Length"$= 0$), it is an ACK frame. Otherwise, it is a DATA frame. Each DATA frame can carry a payload from 0 to MAX (a predefined value) bytes. The MAC destination and source addresses that follow the "Length" field each occupies 2 bytes. The 2-byte filed "Protocol" identifies the upper layer protocol encapsulated in the payload. The fields from "Length" to "Protocol" form the MAC frame header. A two-octet Cyclic Redundancy Check (CRC) over the frame header and payload is appended to the end of payload. The Reed-Solomon correcting code over the MAC header, payload and CRC is added to the end of each DATA frame. A three-octet preamble is appended to the beginning of each frame for synchronization.

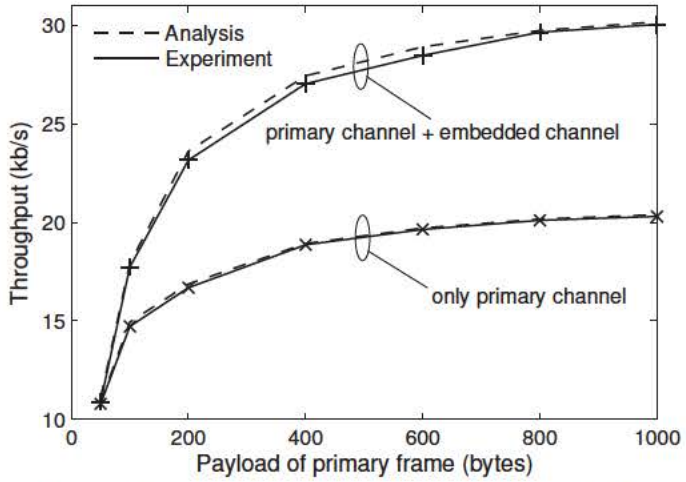
## 3.6. Experimental Evaluation

This section evaluates the proposed protocols through experiments. The Debian Linux System with kernel version 3.8.13 and the Xenomai patch is run within the BBB board. The electronic devices used in our implementation are summarized in Table 3.3. The symbol period is set to 20 $\mu$s for both LOW and HIGH symbols and the (232,200) Reed-Solomon code is adopated at each node. Different to a primary frame, an embedded frame is coded with OOK modulation without the Manchester code (bit "0" is mapped to symbol LOW and bit "1" to symbol HIGH). Note that in the experiments this chapter uses multiple narrow-FOV LEDs to emulate a wider-FOV LED when necessary. All the experiments are carried out in an indoor environment with normal office lights on where the noise level is around 90 lux.[1] The value of `maxNumLOW` introduced in

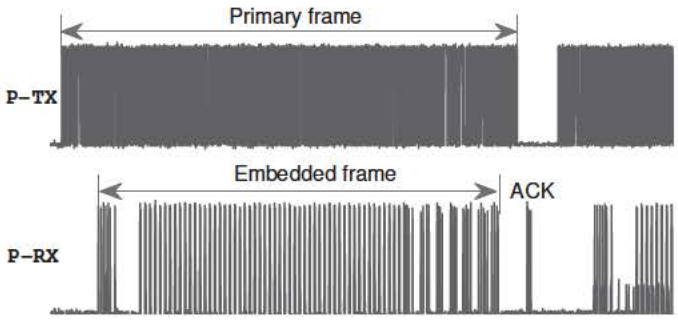
[1] The noise floor is measured by an Android phone (Huawei MATE1).

Table 3.3: ELECTRONIC DEVICES USED IN THE IMPLEMENTATION.

| Model | Description |
|---|---|
| HLMP-EG08-YZ000 | Low-power 5 mm red LED with a Field-Of-View (FOV) of 8° |
| SN74HCT125N | 8-bit buffer with tri-state outputs |
| TLC272CP | Transimpedance operational amplifier |
| MCP3008 | 10-bit analog-to-digital converter |




(a) Saturation throughput. *The primary channel is P-TX→P-RX; the embedded channel is P-RX→P-TX.*




(b) Oscilloscope snapshot *(the primary/embedded/ACK frames shown here are transmitted by their corresponding nodes, not received).*

Figure 3.9: Evaluation results of a point-to-point link.

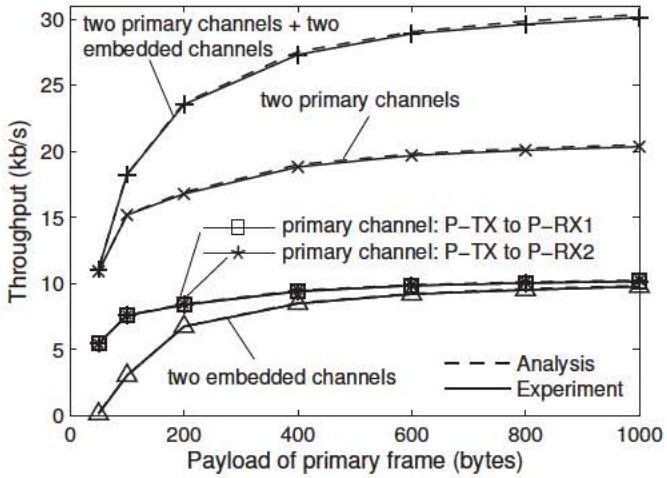
Algorithm 2 is initiated to 20. Besides, for simplicity and unless otherwise specified, the bits "0" and "1" in the experiments are set to be evenly distributed within the payload of frames (i.e., the percentage of bit "1" in the payload is 50%).

### 3.6.1. Performance with embedded channels

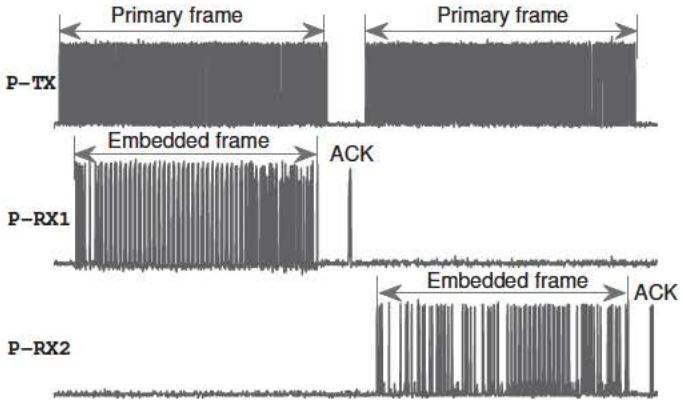
**Point-to-point link.** This subsection evaluates the MAC layer saturation throughput of a point-to-point link without and with the intra-frame bidirectional transmissions, where the nodes are within the FOV of each other. The saturation throughput is achieved under the setting that nodes always have data to transmit. Throughput versus the per-primary-frame payload is shown in Figure 3.9(a), where the payload varies from 50 to 1000 bytes. First it can be observed that the results from the experiments and the analysis match with each other very well. As expected, the throughputs under both protocols in consistent with the payload, *e.g.*, with the intra-frame bidirectional transmission, the saturation throughput is around 11 kb/s when the payload is 50 bytes, while the throughput achieves a value of 30 kb/s when the payload increases to 1000 bytes. Another observation from Figure 3.9(a) is that the performance with embedded channel outperforms greatly the performance without embedded channel when the payload is long, *e.g.*, when the payload is 1000 bytes, the throughput of the former is about 150% of that of the latter. This achievement comes from the intra-frame bidirectional transmission. The throughputs of the two protocols are almost the same when the payload is 50 bytes. This is because the transmission of an effective embedded frame is nearly impossible when the payload of a primary frame is too short. An oscilloscope snapshot of the protocol with embedded channel is shown in Figure 3.9(b), where it can be sees clearly the interaction of nodes as well as the transmission of embedded frames.

**Three-node network.** In this scenario, one node acts a base station and keeps sending data alternately to the other two nodes (*i.e.*, the base station uses the round-robin scheduling algorithm). Again, it can be observed from Figure 3.10(a) that the result from experiments matches well with that from analysis. As expected, it can be seen from the figure that the throughputs at the two receivers are similar under various frame payloads. Also, the throughput gain of the protocol with the embedded channel over the protocol without it increases with the frame payload. This gain achieves a value of $50\%$ when the payload is $1000$ bytes. The interactions between these nodes are illustrated by the oscilloscope snapshot in Figure 3.10(b).

**Four-node network.** The performance of a four-node network is also evaluated. The settings can be referred to the left subfigure of Figure 3.1, where the four nodes A, B, C, and D act as P-TX, P-RX, S-TX, and S-RX, respectively. The oscilloscope snapshot of the interactions between these nodes is presented in Figure 3.11(b). It can seen clearly that three intra-frame concurrent transmissions are obtained. The corresponding saturation throughputs of this network are presented in Figure 3.11(a). It can be observed that, for both protocols, the throughput increases with the payload size. The protocol with embedded channels starts to outperform greatly the protocol without embedded channels after the payload is larger than $50$ bytes. The maximal throughput gain is around $100\%$ after the payload reaches $1000$ bytes, twice as large as the gains of the point-to-point link and three-node network. The reason for this is that, in the four-node network, an additional intra-frame transmission is enabled between the S-TX and S-RX.




(a) Saturation throughput. *The two primary channels are P-TX→P-RX1 and P-TX→P-RX2; the two embedded channels are P-RX1→P-TX and P-RX2→P-TX.*



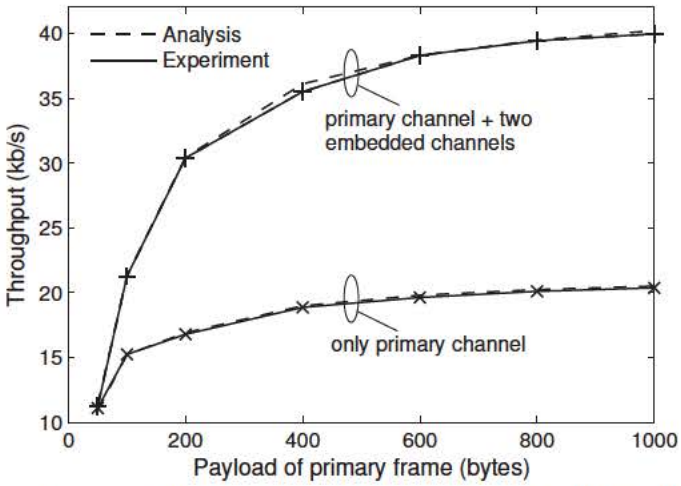

(b) Oscilloscope snapshot *(primary/embedded/ACK frames shown here are transmitted by their corresponding nodes, not received).*

Figure 3.10: Evaluation results of a three-node network.

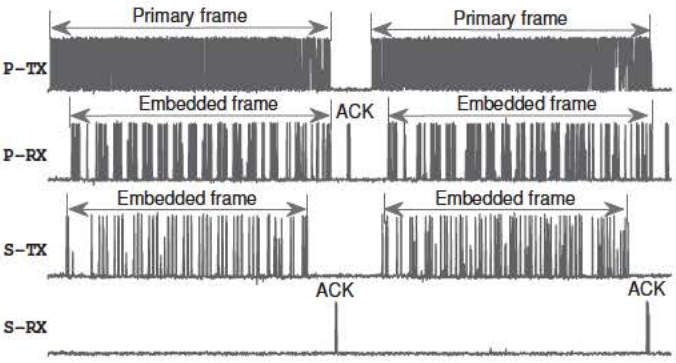
### 3.6.2. Ad-hoc scenario

This part evaluates the proposed CSMA/CD-HA protocol through experiments. As in the numerical results presented in Sec.**??**, this subsection compares the performance of CSMA/CD-HA with those of CSMA/CA and CSMA/CD. In all the protocol, an 8-symbol period is used for channel sensing, *i.e.*, to determine whether the channel is clear or not. The minimum and maximum value of the contention window are set to 4 and 32, respectively. The maximum frame retransmission times is set to 4. Both CSMA/CD and CSMA/CD-HA uses the same backoff mechanism as in the traditional CSMA/CA protocol. While transmitting, a node adopting CSMA/CD or CSMA/CD-HA stops the ongoing transmission if a collision is detected.

**Three-node network without hidden nodes.** The saturation throughput in a three-node net-




(a) Saturation throughput. *The primary channel is P-TX→P-RX; the two embedded channels are P-RX→P-TX and S-TX→S-RX.*




(b) Oscilloscope snapshot *(primary/embedded/ACK frames shown here are transmitted by their corresponding nodes, not received).*

Figure 3.11: Evaluation results of a four-node network.

work without hidden nodes is shown in Figure 3.13. The payload is set to vary from 50 to 1000 bytes. As in Section **??**, it can be observed that the throughputs with different protocols increase consistently with the payload, and the experimental results match well with the analytical results. The CSMA/CD is observed to outperform CSMA/CA, as shown in [37]. The throughput with CSMA/CD-HA outperforms that with CSMA/CD, due to the enabled embedded channel. Both of them can achieve higher throughputs than CSMA/CA, because they can detect collisions during the frame transmission.

**Networks with different number of nodes.** The saturation throughput under different number of nodes is shown in Figure 3.12, where nodes varies from 2 to 4, and payload changes between 200 and 800 bytes. First it can be observed that results from experiments and analysis match well with each other. It can also be noticed that the throughput with CSMA/CA decreases as the number of nodes increases, while the throughputs with the CSMA/CD and CSMA/CD-

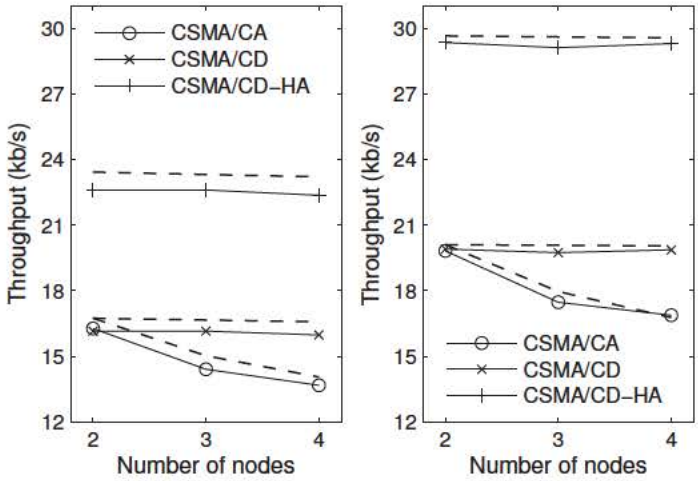



Figure 3.12: Evaluation results of an ad-hoc network with different payloads of primary frames and different number of nodes (*corresponding analytical results are represented by the dash lines*).

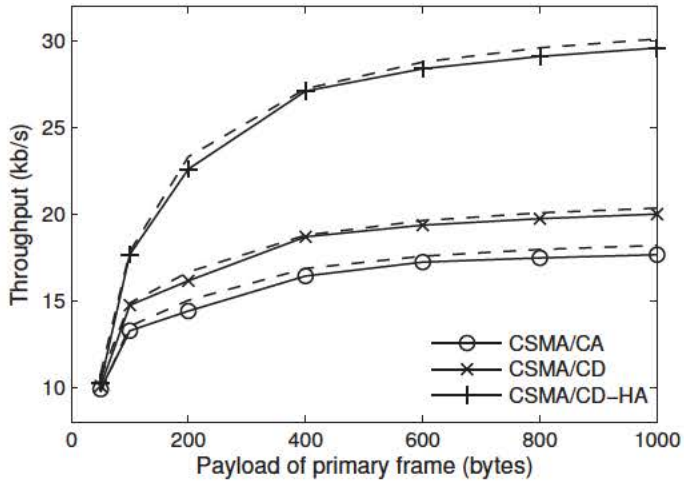
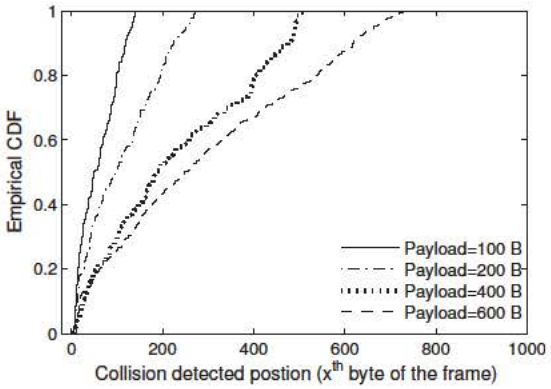


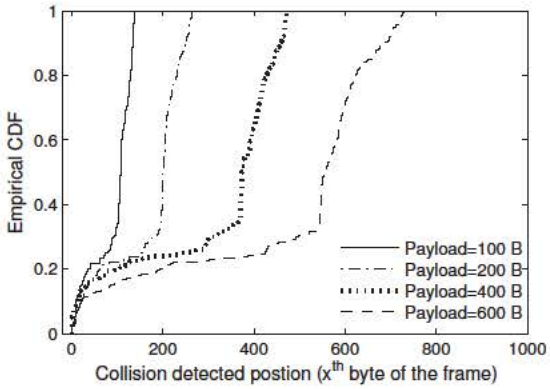

Figure 3.13: Evaluation results of a three-node network (*corresponding analytical results are represented by the dash lines*).

HA are almost not affected. Besides, it can be seen clearly from the results that CSMA/CD-HA can achieve much higher throughput than CSMA/CA and CSMA/CD, independent of the number of nodes and payload length. For example, in Figure 3.12(b) and when there are 4 nodes, the throughput gains of CSMA/CD-HA over CSMA/CA and CSMA/CD are around 70% and 50%, respectively.

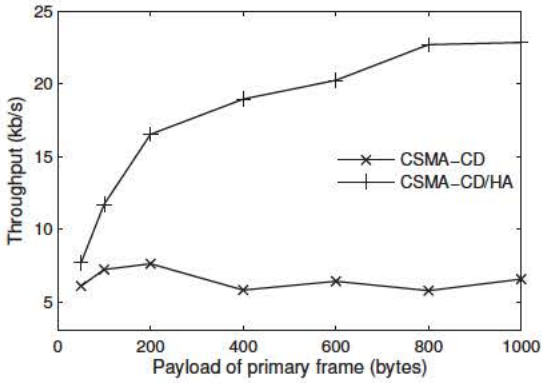
**Three-node network with hidden nodes.** This subsection also evaluates a three-node network with hidden nodes, similar to the illustration shown in the right sub-figure of Figure 3.1. The experiments are carried out in non-saturation scenarios. Under a scenario with specific-length frames, the inter-frame interval (interval between a node finishes transmitting a frame and




(a) Collision position of CSMA/CD




(b) Collision position of CSMA/CD-HA




(c) Non-saturation throughput

Figure 3.14: Evaluation results of the three-node network with hidden nodes.

that node has a new frame for transmission) is set to a value doubles the time taken to transit one of that specific-length frames at the PHY layer. This part compares the CSMA/CD-HA with the CSMA/CD protocol. The position where a collision is detected within each collided frame is presented in Figure 3.14. It can be observed from Figure 3.14(a) that under CSMA/CD, the collisions occurs at positions that are almost evenly distributed within the frame. This is because the two TXs can not hear each other's transmission. In turn, this can cause a collision at any part of the ongoing transmitting frames. However, the collisions under CSMA/CD-HA occur either at the beginning or at the end of collided frames, as shown in Figure 3.14(b). This is due to the fact that a RX starts sending an embedded frame after it receives the source address of the corresponding primary frame, and finishes sending the embedded frame before the whole primary frame is fully transmitted (as illustrated in Figure 3.4). Although there are still collisions caused by the hidden node under CSMA/CD-HA, the number of collisions is actually reduced greatly. This is shown in Figure 3.14(c), where the saturation throughput versus frame payload is presented.It can be observed that CSMA/CD-HA always outperforms CSMA/CD greatly independent of the frame payloads, and the throughput gain can be up to three times higher when the frame payload is set to 1000 bytes.

### 3.6.3. Bit scrambling

The performance of the intra-frame bidirectional transmission can be greatly weakened if:

- the payload of a primary frame has two many "1", or
- the embedded frame payload has long sequences of bit "0"

In the first case, the payload of an embedded frame would be shorten significantly. In the second case, the system would be less resilient to collisions and hidden nodes. Therefore the P-TX may stop transmitting the ongoing primary frame. To solve these problems, this work adopts the *bit scrambling* technique over the payloads of both primary and embedded frames. This work chooses bit scrambling because it has the ability to convert any sequences into seemingly random sequences, thus avoiding long sequences of bits of the same value [61].

Let us use a 16-bit Fibonacci Linear-Feedback Shift Register (LFSR) to implement the bit scrambling. The feedback taps are at the 16th, 14th, 13th, and 11th bits, *i.e.*, the feedback polynomial is $x^{16} + x^{14} + x^{13} + x^{11} + 1$. The nonzero state `0xACD1U` is chosen as a start state of the 16-bit Fibonacci LFSR. The experiments are carried out on a point-to-point link. The payload of primary frames is fixed to 800 bytes, while the percentage of bit "1" within payload varies within the range $[1/8, 1/4, ..., 3/4, 7/8]$. The performances under three different cases are compared: $i$) no embedded channel; $ii$) one embedded channel; $iii$) bit scrambling. In the last case, a dynamic scrambling approach is used. The scrambler in the primary frame is enabled only if the percentage of bit "1" within the payload is above $50\%$. The idea of this approach is to dynamically take advantage of longer set of bit "1"s to increase the system throughput. To implement this, a bit of the SFD is used to inform the receiver whether the scrambler is enabled or not in the current frame.

The evaluation results are shown in Figure 3.15. As previously, it can be noticed that the experimental results match well with the analytical results. As expected, the throughput with no embedded channel is not sensitive to the percentage of bit "1" at all. However, the throughput with one embedded channel depends heavily on the percentage of bit "1", *e.g.,* when $1/8$ of the bits within the payload are "1", the throughput can reach up to 36 kb/s; while it drops to 23 kb/s when the percentage of bit "1" becomes $7/8$. Due to the bit scrambling's ability of converting any sequences to seemingly random sequences, the throughput with bit scrambling can keep stable around 30 kb/s after the percentage of bit "1" in the payload exceeds $50\%$.

## 3.7. Summary

This chapter introduced the intra-frame bidirectional transmission approach for visible light communication networks using one "optical antenna" for transmission and reception. Based on it, this work presented the design, analysis, implementation, and performance evaluation of the

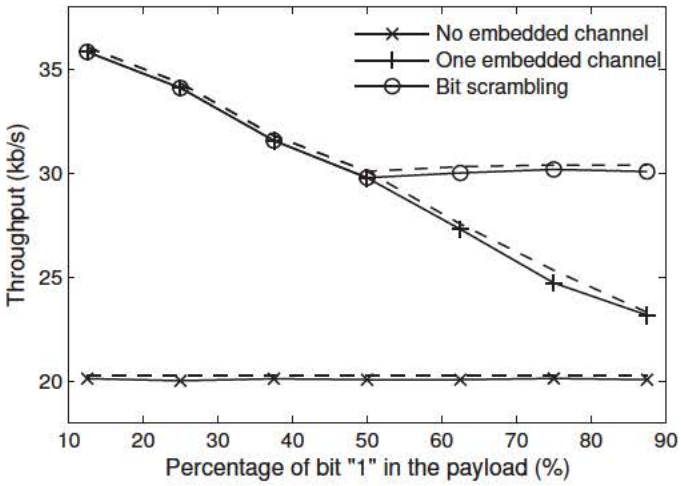



Figure 3.15: Evaluation results of a point-to-point link with bit scrambling (*corresponding analytical results are represented by the dash lines*).

CSMA/CD-HA MAC protocol. This chapter implemented the protocol in a software-defined embedded platform, solved practical challenges, and showed its superior ability to detect collisions and alleviate hidden nodes, thus boosting the system throughput.

# Chapter 4

# Reliable Link for the Internet of Lights

## 4.1. Introduction

Wireless communication based on the radio-frequency spectrum has revolutionized the way our societies work. After decades of research, the radio link is so-well understood that people are now able to embed tiny wireless transceivers on any *'thing'*. These great advances are enabling us to build the so-called Internet of Things (IoT). Exploiting the visible light spectrum could provide equally disruptive effects. This chapter takes one of the first steps in analyzing the complexity of visible-light links from a *networking* perspective. It exposes some of the challenges faced by visible light links and propose initial solutions to overcome them.

**The case for an Internet of Lights.** Wireless communication with visible light is an idea that dates back to the $19^{th}$ century. This old concept is having a strong come back due to two reasons. First, thanks to advancements in visible light communications (VLC), LED lights can now be modulated at high speeds (Kb/s and Mb/s) making them competitive alternatives to wireless communication. Second, due to the high energy efficiency of LEDs, any device that emits light nowadays is likely to be LED-based: car lights, city lights, billboards, toys, wearables and home appliances are already LED-based, just to name a few. Thus, in the future our society could have a new type of pervasive infrastructure waiting to be networked, an Internet of Lights (IoL) [1]. IoL will integrate communication, sensors and light, and create new pervasive smart environments for connected devices and objects, all centered around the light as a medium of communication.

**The challenges.** Achieving the vision of an Internet of Lights requires investigating a new network stack to exploit and adapt to the unique properties of visible light signals (compared to radio). A first necessary step towards a new network stack is to have a reliable visible light link. Designing such a link however is a difficult task due to three main challenges.

*Challenge 1: Limited range and receiver saturation.* Light has a dual nature - it exists both as wave and as photon. As a wave, it obeys the free space path loss model, where the decay of the signal power with the carrier frequency $f_c$ can be expressed as $20 \log_{10} f_c$. As such, the path

loss of visible light signals is $10^{10}$ (70 dB) times greater than that of micro-wave radios. This leads to much shorter ranges for the same output power, a fundamental property that can not be avoided. But visible light signals are also affected by its photonic nature: changes in illumination levels (photons) will increase the noise floor, and the induced noise can have a drastic impact. For instance, a variation from outdoor illumination in a tunnel ($75\,\mu$W/cm$^2$) to outdoor sunlight ($1.80\,m$W/cm$^2$) causes an increase in the noise floor of up to 24 dB. The drastic changes in the noise floor in VLC systems can saturate optical receivers, making optical links disappear.

*Challenge 2: Unstable links due to directional coverage.* The vision for the IoL is not to deploy lights for the sole purpose of communication – as explained above radio is more efficient –, but to piggyback communication on LED lights used for the purpose of illumination. While a light source can in principle be omnidirectional, most LED lights are highly directional. Directional sources are good for avoiding collisions and create spectrum reuse opportunities, but undesired for discovering and maintaining links in mobile settings. Radio mm-wave communication also faces similar problems due to directional coverage, and it is solved partially by the use of antenna arrays than can change the direction of beams. LEDs however cannot direct beams in different directions as in mm-wave because the wavelength of visible light is comparable to or below the size of molecules. As such, smart antenna devices with dimensions in the order of $\lambda/2$ cannot be constructed. In addition, the constraint of constant illumination for the human eyes refrains from beaming in specific directions. Thus, the problem of directionality is more acute in visible light links than in radio.

*Challenge 3: Multiple types of antennas.* Contrary to radio frequency systems, where all nodes use the same transceiver, the transmitter in VLC is always an LED, but different optical receivers can be used each with unique characteristics: photodiodes, cameras and even LEDs can operate as photodetectors, possibly integrated in the same consumer device. Consequently, a network stack for the Internet of Lights should consider not only the fact that light has different propagation properties than radio (Challenges 1 and 2), but also that multiple types of optical antennas could be available at the receiver.

**Contributions.** Considering the above described challenges, this work provides three main contributions towards increasing the reliability of visible light links.

- This work identifies the need for a multi-antenna receiver and provide a deep analysis about the tradeoffs that need to be consider in its design. It shows that LEDs and Photodiodes have complementary properties in terms of transmission range, saturation point and directionality when operating as photodetectors. It exploits the advantages of each type of photodetector and consider their shortcoming using a holistic approach. This leads us to a communication link that is more robust to mobility and more resilient to various illumination levels, from complete darkness to daylight. [Section 4.3]

- Based on experimental insights, this work designs, implements and evaluates a flexi-

ble platform and network stack, from the hardware to PHY and MAC. Our platform is built largely enhancing the open source OpenVLC project, it can sense the surrounding level of illuminance and switch seamlessly among different receiver configurations within the same frame. [Section 4.4]

- This work showcases the applicability of our findings on a scaled down application, where two mobile nodes try to maintain a link under different types of paths, straight and curves, and under different illumination conditions, day and night. In this use-case scenario, our results show that our multi-antenna receiver is able to maintain a constant link while a classical receiver can only offer intermittent connectivity. [Section 4.5]

## 4.2. Related Work

Visible light communication is a very active research area. This section focuses on works that are the most relevant for this contribution. For a complete and recent survey of research in visible light-based networks and sensing please refer to [**?** ] and [62], which capture the strong attention that VLC is receiving from both academia and industry.

**High-end platform for static scenarios.** Researchers have investigated VLC for the next generation of cellular networks [33, 45] and as part of the 802.15.7 standard [28] for short-range communication with visible light and its amendment (currently under preparation). Most of this research area is based on resource-rich platform, such as the USRP and WARP software-defined radios [35, 56, 63, 64]. More recently, interest has spawn in the areas of vehicle-to-vehicle communication [4, 31] and screen-to-camera communications [50, 51]. These works aim at achieving high data rates in *static point-to-point scenarios under mild lighting conditions*, such as LiFi. Efforts beyond these type of scenarios are still in their infancy [**?** ]. Furthermore, the complexity and implementation costs of these platforms are much higher than our proposed system.

**Applications for Internet of Things.** Another category on VLC research is based on low-cost solutions for IoT applications. Visible light sensing for human motions [65], mobile interaction [66] and indoor localization [3, 47] have recently emerged as IoT applications using low-end platforms. These resource-constrained platforms provide low throughputs, for example a 1 kb/s data rate is obtained in [67] to provide a low-cost light bulb for IoT applications. In general, low-end platforms sit at the opposite extreme of the resource-rich high-throughput systems that initiated the investigation of VLC. Our platform is also a low-end system, but our contribution differs from previous work on the fact that their design, in particular the selection of the receiver, was made to suit the needs of a particular application scenario. This work focuses on VLC systems with a more flexible and reliable Data Link Layer to tackle the challenges of dynamic ambient light and mobility conditions.

**The problem of a reliable link.** The motivation of this work is that, in contrast to most radio channels, the gain of the visible light channel is highly dependent on the alignment between transmitter and receiver, as well on the saturation of the receiver, which calls for novel methods

and algorithms. Complementary to this work, [56] studied the problem of adapting the PHY rate within the same frame for photodiodes, which becomes necessary in mobile scenarios with SNRs that drastically change with small movements of the receiver. On the other hand, as pointed out in survey [**?** ], using LED as a receiver has been largely ignored, except for a few works that solely focus on low-power LED-to-LED networks [37, 68]. All of the above studies focus on a single receiver, which are only a subset of the possible configurations of our system. None of the works in the literature did provide evidence of the complementary features of photodiodes and LEDs in order to achieve a stable communication link.

**Research platform.** The author of this thesis proposed the initial OpenVLC platform in [52, 69], targeting low-power LED-to-LED communication. After that, the platform is extended with Photodiodes and high-power LEDs, and the problem of the saturation of the photodiode is reported in [**?** ]. This works extends the prior efforts in three main ways. First, the sensitivity analysis of the photodiode for various values of $R$ and the tradeoff analysis for PDs and LEDs based on illumination and directionality (Section 4.3). Our initial work used only the default resistor for the PD and focus only on illumination not directionality. Second, the final implementation of the software stack, in particular the dynamic setting of the photodiode based on the noise floor and the parallel processing of PD and LED streams (Section 4.4). Third, in the evaluation, only Section 4.1 was presented before, Section 4.2 was redone to consider directionality and Section 4.3 is new.

## 4.3. The question of pervasive communication with visible links

This section first provides the essential background to understand the concept of bandwidth in VLC, and then introduces and experimentally characterize the different antenna receivers available with visible light links. Motivated by the experimental insights, this work then describes the two macro challenges faced by visible light links: high illumination and mobility.

### 4.3.1. Understanding the Bandwidth in VLC

To understand some of the key phenomena affecting the performance of VLC links, there is a need to distinguish between the *optical bandwidth* of the signal in the optical spectrum against the *electrical bandwidth* of the baseband signal:

- The optical bandwidth describes the width of the visible light signal in the optical medium. Having LEDs as emitting technology, this optical bandwidth changes depending on the color used, and is up to a few hundreds of THz (for white light). The optical bandwidth of the photodetector can vary greatly, from very narrow for a particular color, to very wide including not only the visible light spectrum but also infrared.

- The electrical bandwidth describes the bandwidth at which the power of the optical signal can be modulated at the transmitter (with intensity modulation, IM) or decoded at the

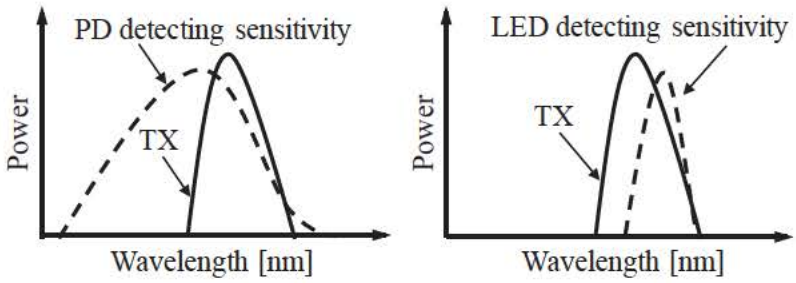
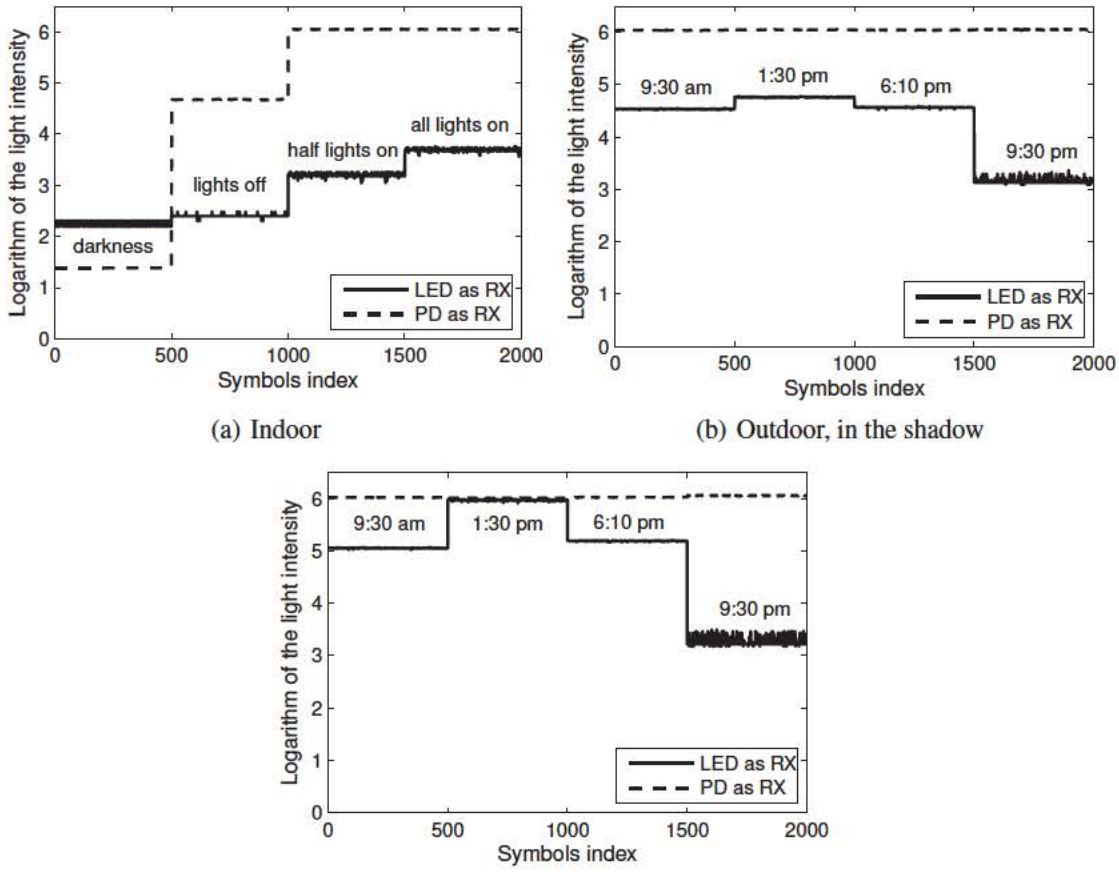



Figure 4.1: VLC spectrum - example of inefficiency in the optical communication.




(a) Indoor

(b) Outdoor, in the shadow

(c) Outdoor, in the sun

Figure 4.2: Measurements of the ambient noise under different scenarios through the LL and PD. The outdoor experiments were carried out on a sunny day of the summer in Madrid, with sunrise at 6:47 am and sunset at 9:38 pm.

receiver (with direct detection, DD). In our work, the electrical bandwidth is determined by OOK modulation, a widely used method for resource-constrained VLC platforms.

The optical bandwidth at the transmitter and receiver determine the efficiency of the visible light link. Let us denote the optical bandwidth as a function of the wavelength $\lambda$ as follows: $0 < S_{\mathrm{TX}}(\lambda) < 1$ (transmitter) and $0 < S_{\mathrm{RX}}(\lambda) < 1$ (receiver). As shown in [62], the overall

optical efficiency of the visible light link can be formulated as $\gamma = \int_0^\infty S_{\mathrm{TX}}(\lambda) S_{\mathrm{RX}}(\lambda) d\lambda$. The maximum efficiency is achieved when the two optical bandwidths match exactly. Smaller $\gamma$ is instead due to a mismatch. This matching effect is depicted in Figure 4.1, where it can be observed that using the LED as transmitter and receiver (right figure) has a better optical match than using the photodiode as the receiver with the same LED as transmitter (left figure).

The final system bandwidth is determined by a number of factors, both in the electrical and optical domain, such as sensitivity, field of view (FoV), resistance to noise, etc. In general, it is intuitive and straightforward to transmit data using VLC, the LED 'simply' has to be turned on and off rapidly. *The communication problems emerge at the receiver side*. There are mainly three types of antenna receivers that are largely adopted in VLC: photodiodes (PDs) [28, 33, 47, 56, 68], LEDs [37, 73**?** ] and cameras [49, 51, 70]. Table 4.1 summarizes the properties of the different receivers. Cameras require significantly more complex hardware and software, and provide a very low data rate, in the order of a few tens of Bp/s. The reminder of this work focuses only on photodiodes and LEDs.

**Photodiodes.** These are the most widely used receivers. Their high sensitivity and bandwidth enables them to achieve high data rates –up to Gb/s– and longer ranges than LEDs and cameras. A photodiode in chip form also has a wide field-of-view, up to $180^o$ (smaller in reality due to the package and lens, and the light intensity required for decoding, $60 - 75^o$). This field-of-view provides a reasonable trade off between being narrow enough to avoid interference but also wide enough to make the link relatively stable to mobility.

**LEDs.** This is the most power- and cost-efficient solution, that uses the same LED for transmission and reception. LEDs use forward bias to emit light: current excites electrons which in turn emit light. In reverse bias, the opposite process occurs: when an LED is off, impinging light excites electrons and generate a current that can be used to decode information.

Each one of these receivers has its own characteristics, advantages and disadvantages, as explained next. The forthcoming tests in this section uses a platform that will be described in detail later. For now, it is only important to know that this platform has three optical elements: a PD, a low power LED (LL) and a high power LED (HL), shown in Fig 4.3. The high and low power LEDs can be used as transmitters, and the low power LED and the photodiode can be used

Table 4.1: Comparison of different VLC antennas.

| | **PD** | **LED** | **Camera** | **Ideal** |
|---|---|---|---|---|
| **Sensitivity** | High | **Low** | Med | High |
| **Electrical bandwidth** | High | Med | **Low** | High |
| **Field of view, FoV** | Wide | **Narrow** | Wide | System dependent |
| **Resistance to noise** | **Low** | High | High | High |
| **Energy efficiency** | High | High | **Low** | High |
| **Bidirectionality** | No | Yes | No | Yes |
| **Cost&Complexity** | Low | Low | **High** | Low |

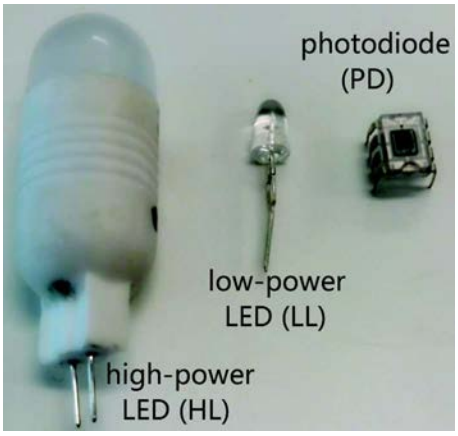



Figure 4.3: Various optical elements used in out platform.

as receivers.

### 4.3.2. The problems of high illumination and mobility

**Limitation of photodiodes: saturation.** Our first goal is to quantify the fundamental limits of a photodiode acting as a receiver. In this regard, two platforms are deployed, $A$ and $B$, located at fix positions 2 m apart from each other with direct line of alignment. The HL of platform $A$ is turned on and both the PD and LL from platform $B$ are used to measure the received light intensity. There is no data communication during these tests. This setup in three different scenarios is deployed: indoor, outdoor in the shadow, and outdoor in the sun.

For the indoor scenario, measurements are carried out during daytime with four different settings: darkness (turn off all light bulbs and block all daylight), lights off (turn off light bulbs, but some daylight is present), half lights on (turn on half of the light bulbs in the room), and all lights on (turn on all light bulbs). In each setting, $500$ data symbols are read from the LL and the PD of platform $B$. The light intensity value for each symbol is measured at the ADC. Denoting $I$ as the output of the ADC value, the y-axis in Figure 4.2 represents $\log_{10}(I)$. If a receiver has a value of $y = 6$, it means it has reached the saturation state. At that point it would be impossible to decode any information.

It can be observed from Figure 4.2(a) that the PD is very sensitive to light changes (dotted line). When half or all the light bulbs in the test room are turned on, the PD reaches its saturation state. The LL, however, is less sensitive to these light intensities (full line). Although the light intensity sensed by the LL increases consistently with ambient light noise, the LL is still unsaturated when all interfering light bulbs are on.

The light intensities sensed by the LL and PD in outdoor scenarios during different times of the day are measured, first in the shadow and then in the sun. For each time of the day, $500$ symbols are read. The results are shown in Figure 4.2(b)-(c). It can be observed that while the PD

is saturated at all times in both scenarios, the LL only gets saturated when exposed to the extreme case of sunlight at the middle of the day, Figure 4.2(c). Overall it can be concluded that the PD alone is not resilient enough to provide a good visible light link.

*Understanding the reason behind the saturation problem.* The PD has a wide optical bandwidth that indiscriminately absorbs a wide spectrum of visible light (as well as infrared frequencies), up to reaching the point of saturation. This optical bandwidth is usually larger than the one of LED transmitters, thus resulting in a low efficiency $\gamma$. In contrast, the higher resilience of LEDs to light intensity is due to two reasons: its narrow FoV and its inherent optical filter. The narrow FoV of LEDs acts as an spatial filter, allowing the LED to absorb energy coming mainly from objects in its direct line-of-alignment, filtering out light sources coming from other angles. The optical filter effect is depicted in Figure 4.1. LEDs are designed to transmit light on a narrow optical bandwidth of the visible light spectrum (less than 30 nm – for comparison the entire visible light spectrum covers approximately 370 nm [34]). When the same LED is used as a receiver –as in the experiment in Figure 4.2–, LEDs have a channel impulse response similar to its spectral emission profile [71, 72], which results in a high efficiency $\gamma$.

**Limitation of the LED: directionality.** As experimentally shown in the previous test, LEDs are good at filtering optical noise for scenarios with high illuminance, both indoors and outdoors. Their main limitation however is their strong directionality, which makes them more susceptible to link instability due to mobility. To depict this problem the following experiment is performed. The position of platform $A$ (transmitter with HL) is fixed and platform $B$ (receiver with LL) is moved around different locations within a sector of 2 m, as shown in Figure 4.4. In this setup, the transmitter emits modulated light, i.e. the receiver can decode packets. It can be observed that the reception range of the LED is narrow in space, around 15 cm $\times$ 200 cm. This would make LED-only links intermittent under mobility. Adopting an array of LEDs in parallel would increase the width of the reception coverage, but it would not be able to provide a continuous FoV comparable to a similar arrangement of photodiodes.

*Design choice*: The most complete optical receiver would have a photodiode, LED as well as a camera. But a close look at Table 4.1 shows that adding a camera would only give us a minor improvement (decode a few bytes per second on a highly illuminated area at a wide angle) for a very high cost (expensive, energy-hungry and complex platform).

While our main goal is to increase the robustness of the optical link, this work is also interested in designing a system that is simple, low-cost and energy-efficient so a wider community could benefit from this work.

### 4.3.3. Tradeoff Analysis

The previous section described in macro terms the problems of saturation and directionality. But to design a robust link, a deep understanding of the tradeoffs is needed that PDs and LEDs have in terms of their reception range, resilience to external illumination and directionality. The analysis of the tradeoffs for the different optical antenna elements will finally allow us to present

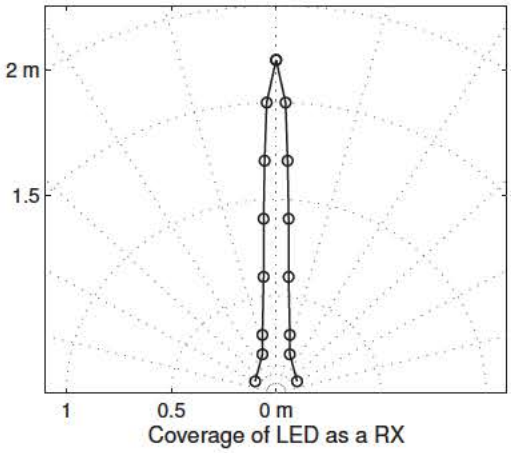



Figure 4.4: Coverage of an LED as a receiver.

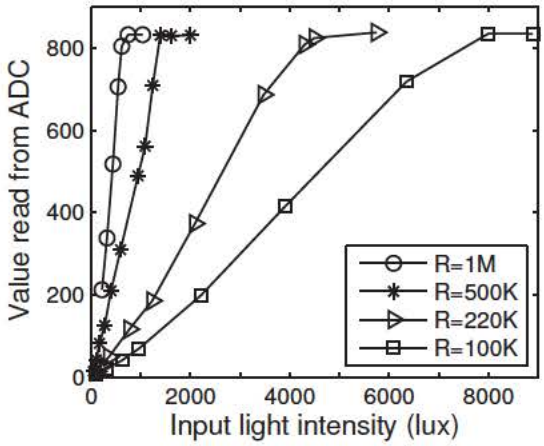



Figure 4.5: ADC output (saturation point) of the PD.

the basic design principles that our Data Link Layer should follow.

**When to use the PD? In dark and mobile scenarios.** The sensitivity of a photodiode depends on the resistor used in its driving low noise amplifier circuitry. Impinging light produces a current that passes through a resistor $R$, and the generated voltage $V$ is read by the ADC. If a large $R$ is used, small changes in current (i.e. small variations in the amount of impinging light) will lead to noticeable changes in voltage, which leads to a very high sensitivity. Thus, a big $R$ enables PDs to have long reception ranges in dark scenarios. But in scenarios with high levels of illuminance, the impinging light, and consequently the generated current, will be large and the ADC will saturate rapidly. A small $R$ has the opposite effect, more impinging light is required to differentiate '1' and '0' bits (less sensitivity), but the PD becomes more resilient to external light sources.

To quantify the relationship between the resistor $R$ and the sensitivity of the photodiode, the ADC values in the platform under different illuminance conditions and with different resistors are measured. The results are presented in Figure 4.5. With the highest sensitivity, $R = 1\,\text{M}\Omega$, a visible light link would disappear below the illuminance levels required for working spaces (500 Lux). With the lowest sensitivity, $R = 100\,\text{K}\Omega$, the visible light link would disappear way below the illuminance conditions observed on cloudy days (10K Lux). Such exposure to saturation would jeopardize the use of visible light links on many indoor and outdoor environments.

The tradeoff for increasing the resilience to external light sources (small $R$) is a reduction of the reception coverage. To highlight this tradeoff the following experiments are performed. The receiving platform $B$ is placed at position [0,0]. The transmitter platform $A$ sends packets using the high-power LED, and it is moved over various concentric circles around [0,0]. Figure 4.6 shows the results, where it can be observed that reducing the value of the resistor $R$ leads to a significant reduction of the reception coverage, in particular on settings with high illumination.

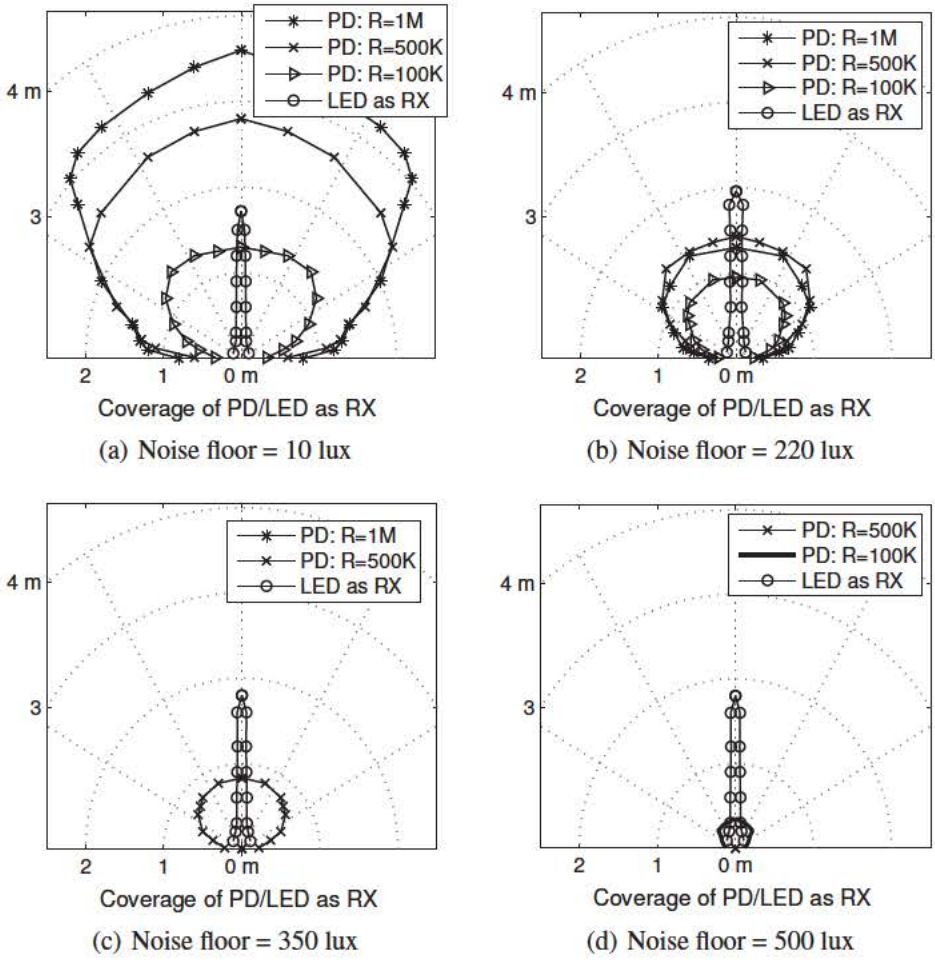



Figure 4.6: Reception coverage of PD and LED for various illumination levels.

Figure 4.7 summarizes the tradeoff between coverage range and resilience to illuminance with a Pareto frontier curve based on the results from Figure 4.6. Note that in general fine-tuning the resistor $R$ is a loosing proposition because tradeoffs lead to an inward curve towards the worst case point [0,0], instead of an outward curve towards the best operational point [max range, max lux]. Overall, our results with the PD lead to the following insight:

*Insight 1: A visible-light link only benefits partially from reducing the resistor $R$ to increase the resilience of photodiodes to higher levels of illumination. This is because the reception coverages achieved with these more resilient settings are very small. As such, the fundamental problem of link saturation can not be solved adjusting the configuration of photodiodes alone.*

**When to use the LED? In static and well illuminated scenarios.** From the previous section, it has been known that varying $R$ is not the solution to a fundamental problem. A static value of $R$ is then considered for the low noise amplifier circuitry of the LED and compare the reception coverage of LED against those of PD. The results are shown in Figure 4.6 for various

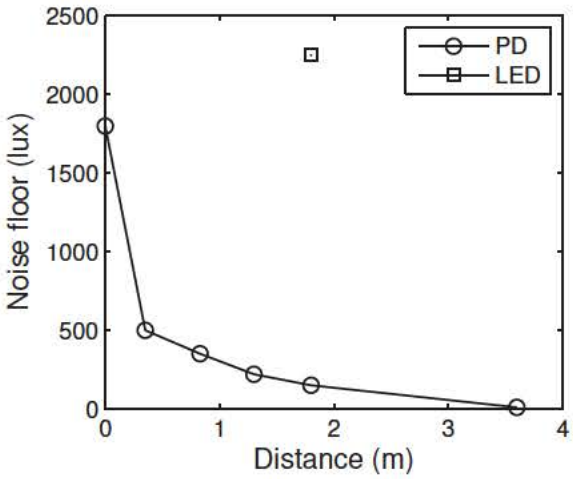


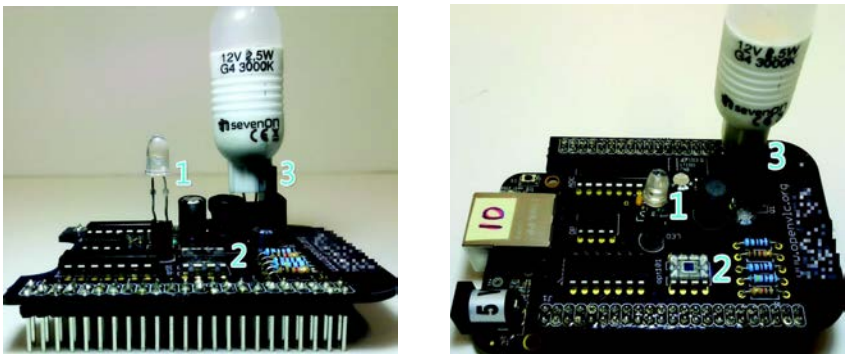

Figure 4.7: Pareto frontier of photodiodes with LED point.

illumination levels. Then three main operational areas can be identified . First, in dark scenarios, Figure 4.6(a), the LED adds nothing to the performance of a visible light link, since its coverage is a small subset of that provided by the photodiode. Second, under mild indoor illumination conditions, Figure 4.6(b)-(c), the LED adds some benefit as a receiver. Third, in normal indoor lighting conditions, Figure 4.6(d), the LED is the only alternative. Note that the coverage of the LED does not change under different illumination conditions. This property is the main advantage of LEDs acting as receivers. Overall, the experiments performed with the LED as a receiver lead to the following insight:

*Insight 2: An LED provides a constant but narrow reception coverage across various illumination conditions. Given the much larger coverage of photodiodes under darker conditions, LEDs should be used for reception only under medium and high illuminance conditions.*

## 4.4. System Design

Ideally this work would like a receiver that works under all lighting conditions (from complete darkness to strong interfering light) and that maintains a wide and long coverage under all these conditions. But as shown in the previous sections, there is no one-size-fits-all solution. The tradeoffs related to coverage, operational range and directionality ask for a multi-layer solution, at HW, PHY and MAC. In order to investigate and tests these trade-offs, a flexible platform is needed, that can provide sufficiently fast reconfigurability in the presence of drastic changes of the illuminance level and/or of high mobility of the user. Our platform is presented in the next subsection.




(a) The OpenVLC cape (b) Cape plugged into the BBB

Figure 4.8: The OpenVLC platform. The embedded board runs a Debian Linux and the OpenVLC driver to interface the OpenVLC cape to the Internet. The optical antennas are: (1) low-power LED; (2) Photodiode (PD); (3) high-power LED.

### 4.4.1. Redesign of OpenVLC Platform

**Starting point of this work.** The system is implemented as a general-purpose software-defined platform OpenVLC. This work started from the initial platform presented in [52, 69], originally targeting low-power LED-to-LED communication. The key approach of [52, 69] – that retained in this work – is to develop a low-end platform that largely relies on open source software for fast prototyping. Data is transmitted using intensity modulation, where binary information is mapped to the presence (symbol HIGH) or absence (symbol LOW) of the visible light carrier. The communication is performed using On-Off Keying (OOK) modulation together with the Manchester code. Based on the received photocurrent, the receiver chain decodes the frame with direct detection as a sequence of modulated symbols and then convert them into binary data.

There are two types of frames: DATA frame and acknowledgment (ACK) frame. Each frame carries a payload up to a predefined MAX bytes. A two-byte Cyclic Redundancy Check (CRC) over the MAC header and frame payload, and the (216,200) Reed-Solomon (RS) error correction code are appended to the end of each frame.

For the implementation, the PHY and Data Link Layers are developed as a new Linux driver that can communicate directly with the VLC hardware and the Linux networking stack. Primitives are also implemented that can be used to develop various PHY and Data Link layer protocols.

**Redesign of the platform** The platform presented above is not sufficient to investigate the challenges introduced in Section 4.1 yet. The platform is largely extended, both in hardware and software. The system is moved from LED-only communication to a system with three optical antennas (high-power LED, low-power LED and photodiode) and provide sufficient flexibility for the reconfiguration. The system architecture is illustrated in Figure 4.9. OpenVLC consists now of three parts:

1) *BeagleBone Black (BBB) board* [1], a low-cost platform equipped with a AM335x 1GHz CPU

[1] `http://beagleboard.org/Products/BeagleBone+Black`

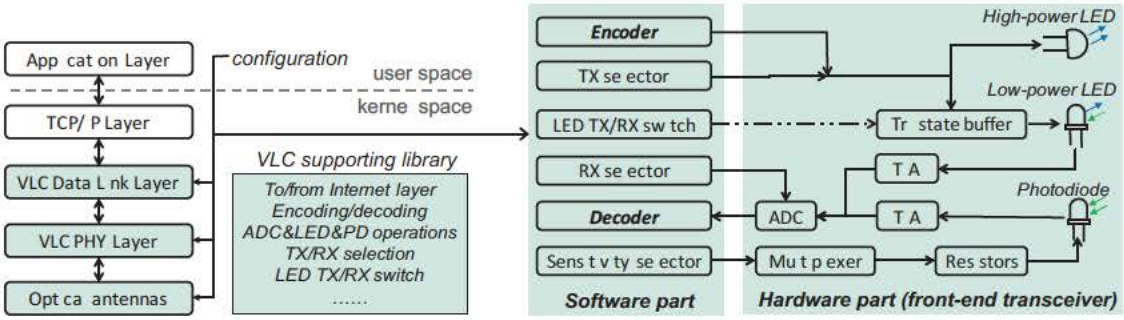



Figure 4.9: System architecture: *left)* system communication stack; *right)* block diagram of the front-end transceiver.

and 65 GPIOs that runs Linux operating system (same board as in [52, 69]).

2) *OpenVLC cape*, a front-end transceiver that can be attached directly to the BBB, as shown in Figure 4.8. The front-end transceiver adopts the following optical antennas (details are in the Table 4.2), connected to ancillary circuitry for transmission and reception:

- a low-power LED (LL) supplied with 3.3 V in both forward and reverse bias. In forward bias, LL requires a current in the order of a few tens of mA.

- a high-power LED (HL) only operating in forward bias. The HL selected in this work consumes 2.5 Watts.

- a PD for light reception supplied with 3.3 V in reverse bias.

Each antenna at the receiver can be connected to a TransImpedance Amplifier (TIA) with gain control. The transceiver can choose an antenna at the receiver (RX) between the LL and PD, and an antenna at the transmitter (TX) between the HL and LL. When the LL is selected as both a TX and RX, it falls back to the original design of LED-to-LED communication presented in [69].

3) *OpenVLC driver*, that implements the software solutions (PHY + Data Link Layer) for VLC networking and seamless reconfiguration of the receiver, as described in the following subsection.

Table 4.2: Electronic devices used in the platform.

| Model | Description |
|---|---|
| HLMP-EG08-YZ000 | Low-power 5 mm red LED |
| Seven On (G4, 3000K) | High-power LED, 12V, 2.5W |
| OPT101 | Photodiode |
| 74HCT244N | 8-bit buffer with tri-state outputs |
| LM358N | Transimpedance amplifier |
| MCP3008 | 10-bit analog-to-digital converter |
| ADG444 | 4-channel multiplexer |

### 4.4.2. Core aspects in the system design

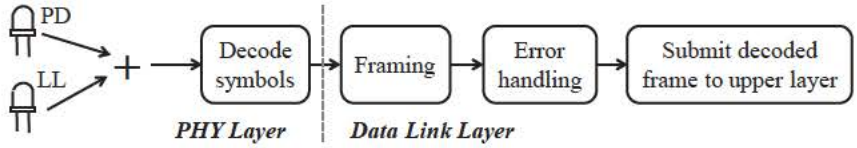
The availability of the platform described in the previous section allows us to set the requirements of our VLC system with the overarching goal of providing a reliable Data Link Layer for the IoL, and then design and test a VLC system that fulfills these requirements. As will see, exploiting visible light links requires challenging much of the know-how inherited from radio systems to re-design the Data Link Layer.

*Requirement 1: The responsibility of checking the channel is both on the transmitter and the receiver*. In radio systems, the transmitter is responsible to schedule the frame and avoid collisions. In visible light systems, scheduling data is challenged by the fact that transmission is a secondary aspect on top of illumination and the "primary receiver" is the human eye. In addition, the directionality makes collision avoidance a second order problem [73]. As such, the receiver has an important role to achieve a reliable communication channel. In order to reach this goal, the receiver will need to first measure the surrounding light intensity (noise floor) and use the most convenient configuration of the optical receiver to decode noisy data. As a consequence of the experimental finding of Figure 4.6, three operational regions are identified:

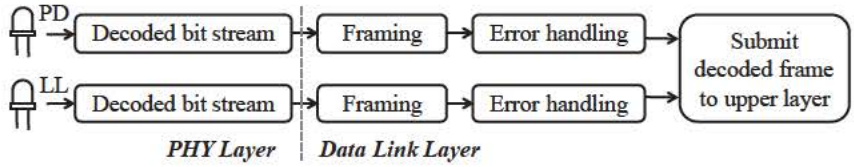
- use only the PD in the receiver for very dark environments ($\leq 100 - 150$ Lux).
- use only the LL in the receiver for normal indoor illuminance and outdoor illuminance from overcast day to full daylight ($\geq 500$ Lux), subjected to the condition that the LL (or any other optical receiver implemented in the platform, in a more general setup) is in the FoV of the transmitting device.
- for illumination levels in between these the two points above, turn on both the PD and LL in the receiver.

*Proposed approach*. This work makes full use of the optical antennas provided by the PHY layer and control in software the TX and the RX antennas. Two potential implementations of the receiver can be distinguished, illustrated in the block schemes in Figure 4.10. In the implementation in Figure 4.10(a), the received signal of PD and LL are summed up symbol by symbol. Then the framing and error handling are performed. The advantage of this approach is that it requires only one receiver chain (a cost effective implementation according to the use case). In Figure 4.10(b) the PD and LL can receive light signals in parallel, with one chain per antenna. The Data Link Layer requests the PHY layer to provide the separated decoded signals (a sequence of bit information) received by the PD and the LL. Then it performs the framing and error handling per each receiver chain (with each chain with one antenna). The Data Link Layer will only forward the successfully decoded frame.

*Requirement 2: The Data Link Layer should be able to read both receiver chains in real time and switch among them fast, even within the same frame*. As stated in the prior guideline, there will be illumination scenarios where both receivers will need to be active. Furthermore, lighting




(a) Summing up the received signal of PD and LL.




(b) Process PD and LL in parallel

Figure 4.10: Receiver implementations to process the received signals from PD and LL.

conditions can change rapidly and significantly. Thus to avoid loosing frames, the Data Link Layer should be able to start decoding a frame with one configuration of the receiver and, if necessary, change the configuration on the fly.

*Our Approach.* The PD's saturation is changed by varying the gain control in the TIA, as the measurements carried out in Figure 4.5. To select the most appropriate gain level, several carefully-selected resistors $R$ are used, connect them to the TIA (see the front-end transceiver in Figure 4.9(b)), and design a software-defined controller to switch the feedback resistor $R$ to the most appropriate one based on the current noise levels and Signal-to-Noise Ratio (SNR). In addition, to convert symbols into binary data and distinguish between HIGH and LOW symbols of the visible light carrier, an adaptive symbol detection threshold is adopted. This is necessary to deal with in-frame drastic variations of signal strength and noise floor level. The detection threshold is first obtained on a per-frame basis by averaging out the digital samples of the preamble sequence. During the reception of one frame, the threshold will be reconfigured if the sensitivity has changed. The new threshold value is recalculated by averaging out a pre-fixed set of the latest received samples. The adaptation of PD's gain control is provided in Algorithm 3. $\text{SNR}_{\text{M2M4}}$ stands for the method used to estimate SNR without measuring the noise, only using the second and fourth moments of the received signal strength.

## 4.5. Performance Evaluation

This section evaluates our Data Link Layer under increasingly complex test cases to quantify the performance of our visible light link under changes in illumination, direction and mobility.

[2]When PD is saturated, the output voltage from the TIA reaches the maximal value. Therefore, the read RSS from ADC is at its maximum (denoted as `maxRSS`). If a certain amount of the raw RSS values in the stream `symStream` reach `maxRSS`, then the PD is declared to be saturated.

**Algorithm 3** The PD's sensitivity adaptation.

**Input:** `symStream`: symbol stream (raw RSS values); `sensLevel`: current sensitivity level; `minSensLevel` / `maxSensLevel`: minimal / maximal supported sensitivity level; `minSNR`: minimal required SNR to decode a frame

**Output:** `sensLevel`: adapted sensitivity level.

```
calculate the SNR_M2M4 from symStream based on [74]
if SNR_M2M4 ¡ minSNR then
    if PD's saturation is detected[2] then
        sensLevel ← sensLevel-1
        return Max(sensLevel, minSensLevel)
    else
        sensLevel ← sensLevel+1
        return Min(sensLevel, maxSensLevel)
    end if
end if
return sensLevel
```

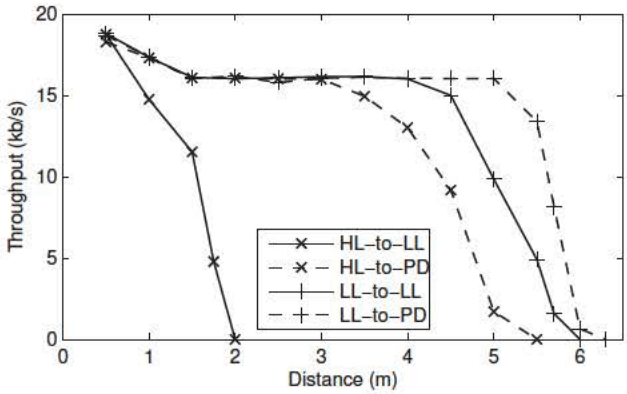



Figure 4.11: MAC layer throughput versus distance under different links (indoor scenario with lights off).

### 4.5.1. The ideal case: static nodes, dark room, line-of-alignment

A simple protocol to quantify the maximum throughput of the platform under ideal conditions is implemented. The transmitter sends a packet and waits for an ACK. If the ACK does not arrive within a predefined threshold, the transmitter sends the packet again. If the packet has a payload (i.e. Length>0), it is inferred to be a DATA frame, else it is assumed to be an ACK. The transmitter sends packets continuously and the receiver is positioned at different distances with direct line-of-alignment. The payload size is 600 bytes.

The experiments are carried out for all TX/RX combinations in a dark room. The results are shown in Figure 4.11. It can be observed that under these ideal conditions using the LL as transmitter or the PD as receiver lead to links with a long range (<4 m) and a high throughput (<15 Kb/s). On the other hand, the tuple HL-LL leads to a weaker link. These outcomes occur

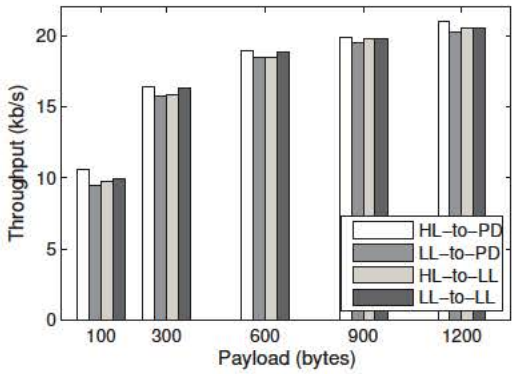


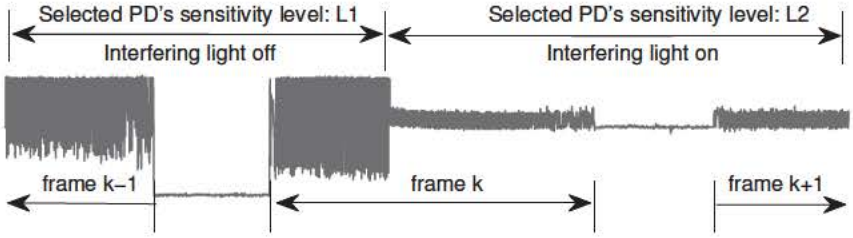

Figure 4.12: MAC layer throughput of different payload sizes (in a dark room).

due to two reasons. First, the LL has a narrow field of view which enables it to focus all its (little) energy on a long distance. Thus, the LL-LL and LL-PD links have long ranges. Second, the HL has a much wider FoV ($120^o$ vs $8^o$ for LL) and hence its bigger power is dissipated over increasingly larger surfaces. This means that HL trades-off shorter coverages for wider ranges. It can be seen however that the reduction of the range is more pronounced when the LL is used as receiver (HL-LL link). This is because the high power LED transmits information with an optical bandwidth that is one order of magnitude larger than LL (approximately 370 nm of bandwidth for HL, the entire visible spectrum, versus 30 nm for LL, around the color red ). Thus, while the PD absorbs most of this energy, making the decoding easier, the LL will filter out a large part of the energy coming from the high power LED. This low absorption leads to a small optical efficiency $\gamma$ and lower generated photo-currents, thus making it harder to decode information. This phenomena is not observed when the low power LED is the transmitter because the transmissions and reception filters in that case are on the same optical bandwidth (coupled), which gives the highest optical efficiency $\gamma$ as explained in Section 4.3.1.

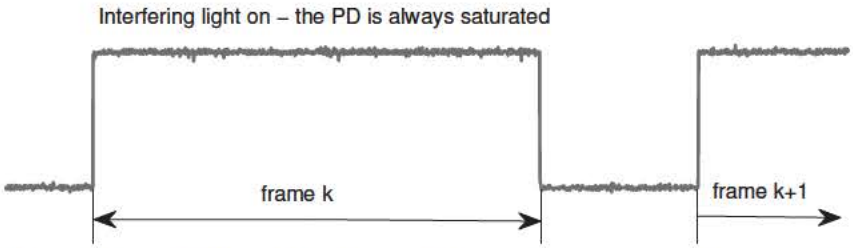
It is important to mention that the maximum throughput depends on the size of the payload. The higher the payload size, the higher the throughput, because the relative overhead of the communication decreases. Figure 4.12 shows some further experiments performed to quantify the throughput as a function of the payload size. In this experiment, the two nodes are located 0.5 m apart from each other. It can be observed that at around 600 bytes the maximum throughput stabilizes. Thus, unless stated otherwise this payload size is used in next experiments. Also, the next experiments only use the HL as a transmitter. This work does this because, even though the HL provides the shorter ranges, it provides the wider coverages and it is a more realistic transmitter since it uses all the visible light spectrum, as most LED lights do.

### 4.5.2. The intermediate case: static nodes, illuminated rooms, no line-of-alignment

The previous subsection analyzed the throughput in a best case scenario: in a dark room and maintaining directionality. Now let us measure the throughput under changes in illumination and




(a) Switch the sensitivities of the PD during the reception of a frame.



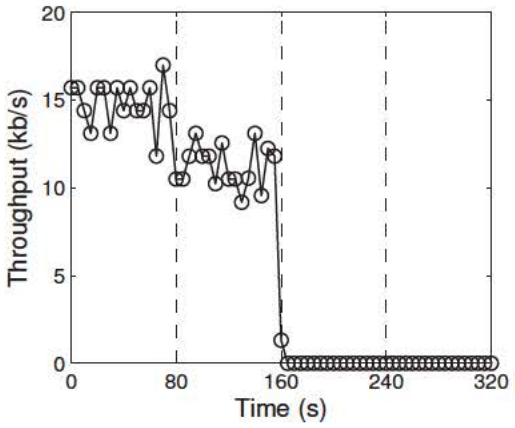

(b) Combine the PD and LL physically during the reception of frames (with interfering lights on).

Figure 4.13: Change the optical transceiver on the fly during the frame transmissions.

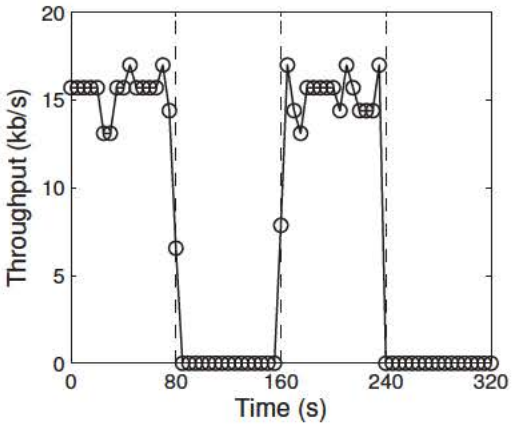
directionality.

First this subsection will describe how our Data Link Layer copes with sudden changes in the environment. The front-end transceiver has software-defined selectors to change the receiver configuration on the fly. This capability is illustrated in Figure 4.13, where an oscilloscope is used to measure the voltage signal at the receiver during frame transmissions. In Fig 4.13(a), within the reception of a frame, the sensitivity of the PD can be automatically adapted based on the current noise floor. Under this case, the frame can still be decoded due to the ability of our decoder to adjust dynamically its thresholds, as explained in Sec. 4.4. Figure 4.13(b) highlights the importance of processing the streams in parallel, as in Figure 4.10(b), as opposed to merging the two raw signals before processing, as in Figure 4.10(a). If the PD is saturated, both signals will end up saturated if merged directly. Thus, this approach would be equivalent to having only the PD.

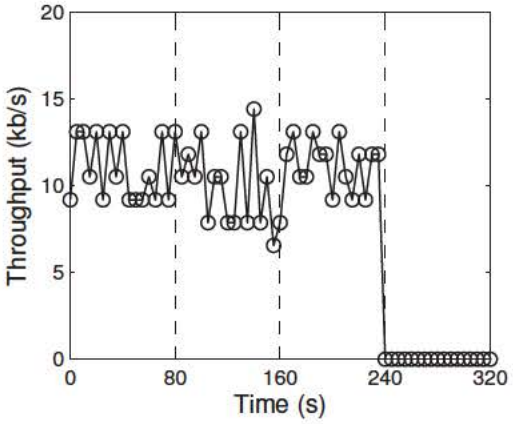
Now let us test the resilience of our system for maintaining a constant throughput in dynamic settings. Contrary to the prior subsection, where the throughput at the Data Link Layer is measured, in these experiments the throughput at the Application Layer is measured. The software of our system is implemented as a Linux driver connected to the TCP/IP layers. Thus the platform's performance can be measured using various traditional network measurement tools. A command that is widely used to measure throughput is `iperf`. In our set up, the IP addresses of the nodes are set to 192.168.0.1 and 192.168.0.2, respectively. Each experiment is run with UDP traffic for 320 seconds, and set `iperf` to report the throughput every 5 seconds. The two nodes are positioned 0.5 m apart and the environment is changed in two ways: by turning on/off the light bulbs in the office and by changing the angle of view between the two nodes. Figure 4.14 shows




(a) Only the LL is the RX.




(b) Only the PD is the RX.




(c) Both PD and LL are the RXs.

Figure 4.14: Evaluation results of the UDP throughput under dynamic environments (HL is the TX).

the results of four different setups.

Setup A: from 0 s to 80 s, angle is $0^o$, lights are off.
Setup B: from 81 s to 160 s: angle is $0^o$, lights are on.
Setup C: from 161 s to 240 s, angle is $45^o$, lights are off.
Setup D: from 241 s to 320 s, angle is $45^o$, lights are on.

Figure 4.14(a), (b) and (c) show the results of using only the LL as a receiver, only the PD as a receiver, and our Data Link Layer. The high directionality of LL suffers as a receiver for wide line-of-view angles, Setups C and D in Figure 4.14(a). The PD on the other hand is resilient to these changes in directionality but it is exposed to saturation effects and the link disappears at high illumination conditions, Setups B and D in Figure 4.14(b) (the PD with the de-facto highest sensitivity level is used). Our Data Link Layer maintains a high throughput in all settings except for the last one, Setup D in Figure 4.14(c), where the high illumination saturates the PD and the

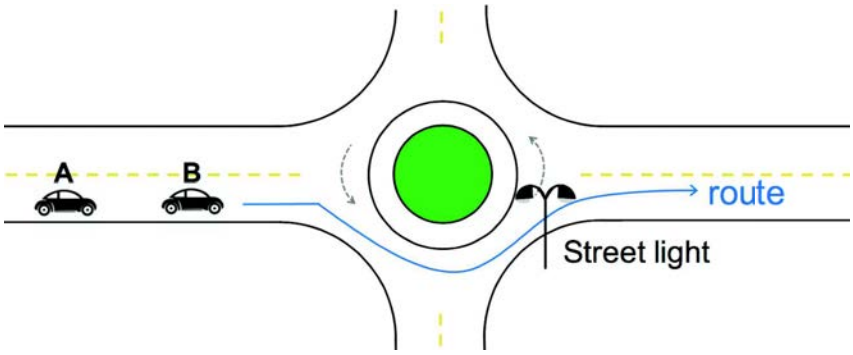



Figure 4.15: Route of the mobile nodes and environment setup for the application test.

wide angle renders the LL ineffective. Note that the throughput of our Data Link Layer is 2-3 Kb/s lower for Setups A,B and C. This occurs due to the extra overhead in incurred by monitoring the noise floor and by processing the two streams of data in parallel.

### 4.5.3. The challenging case: mobile nodes, illuminated rooms, no line of alignment

Until now the resilience of proposed link has been evaluated in scenarios where the nodes do not move. These type of scenarios are of interest because most lights are fixed, such as indoor light in buildings or street lighting. However, other scenarios may have mobile nodes with LED lights, for example, cars and motorbikes. This subsection will show the performance of the link in a scaled down mobile set up.

Two nodes are used in a scenario as shown in Figure 4.15, where node A "chases" node B. At the beginning, the two nodes are put 40 cm apart from each other, and then are moved manually but continuously at approximately the same speed (5 cm/s). The aim is to maintain a reliable link under different types of paths, straight and curves, and under different illumination conditions. The route followed by the two nodes is denoted by the blue line in Figure 4.15, marked as 'route'. Assume node B needs to transmit data continuously to node A. A lamp is placed at the end of the round about to act as a road light. This experiment is performed under two illumination conditions: with all lights in the office off, except for the 'street lamp' (to emulate night conditions) and with all lights on (to emulate daylight conditions). The results are shown in Figure 4.16. Each marker 'x' represents the reception of one frame, and the time in the x-axis is normalized to capture the beginning and end of the route, since the time taken to cover the route at each run is not exactly the same. Horizontal white spaces between any two continuous 'disconnected' markers 'x' imply that the communication link is lost for that amount of time.

Different types of receivers are considered: first the experiments are performed by setting the resistor of the photodiode to a single value (SoloPD-X), the experiments are run by using only the LED (SoloLED), finally the experiments are run with our visible light link (M3(X)). To showcase the adaptability of our link, the received packets based on the photodetector and configuration

used are marked. The label M3(allPD) is the aggregated response obtained from merging the input from the different sensitivity levels of the PD.

In Figure 4.16 it can be observed that at the highest sensitivity, the PD has a poor performance at night when it is in the vicinity of the road lamp (SoloPD-L1 in Figure 4.16(a)), and the link is simply non existent during the day (Figure 4.16(b)). As the sensitivity of the photodiode is decreased (SoloPD-L2 and -L3), the link becomes more resilient, but if longer internode distances would have been used, the links would disappear. The SoloLED configuration performs reasonably well in both scenarios (day and night) except for the area close to the round about where the line of alignment is affected.

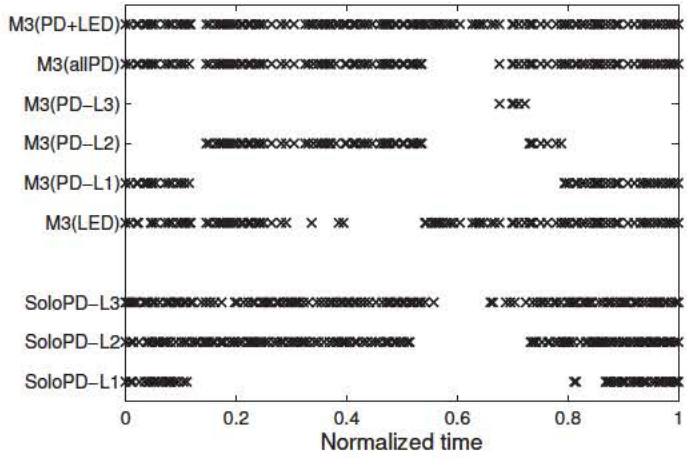
Now let us look into the performance of the two main properties of our Data Link Layer: the ability to adjust the resistor of the PD based on the surrounding noise floor, and the parallel processing of streams coming from the PD and LL. The performance of the first property is captured by the traces M3(PD-L$X$) and the aggregated M3(allPD). For both scenarios, day and night, our Data Link Layer protocol is able to select the right PD sensitivity and leads to reception rates that are the same as those as having three PDs running in parallel with different sensitivities. The performance of the second property, merging PD and LL streams, is captured by the trace M3(PD+LED) which is the only configuration that provides constant connectivity for this setup.

## 4.6. Discussion

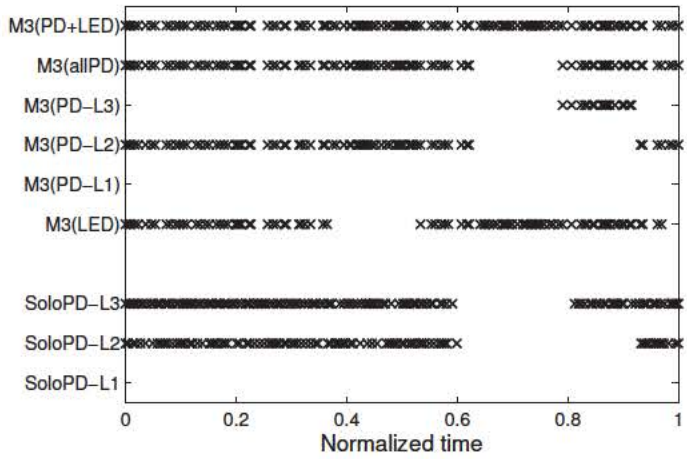
This work investigates the initial concepts and solutions to a critical problem to bring VLC systems into reality: how to provide a reliable visible light link. This section exposes the limitations of our current work and make a step ahead and discuss the potential research directions brought by our findings.

### 4.6.1. In-frame dynamic receiver selection

This work has proposed to use the two receivers PD and LL in parallel (Sec. 4.4) and evaluates its performance (Sec. 4.5). It has been shown that this approach provides better link connectivity by adding processing overhead and sacrificing the throughput. To reduce the overhead and recover the throughput, one solution is to always select the best receiver within the reception of a frame. Our system already has the ability to switch the receivers on symbol-level. Therefore, an efficient algorithm that can swiftly switch between different antennas within a frame (*note that one antenna's - the PD's - sensitivity has been already switched on symbol-level in current system, as shown in Figure 4.13(a)*) should be able to reduce the overhead on parallel processing and bring a stable decoding. Even though different algorithms may cause the wrong reception of some symbols during the symbol-level switching of different antennas, the RS error correcting code implemented in the system can erasure these wrong symbols to some extent (depending on the configuration of the RS code). The main problem with in-frame dynamic receiver switching is




(a) All office lights are off, except for the 'street lamp' (to emulate night conditions)




(b) All office lights as well as the 'street lamp' are on (to emulate daylight conditions)

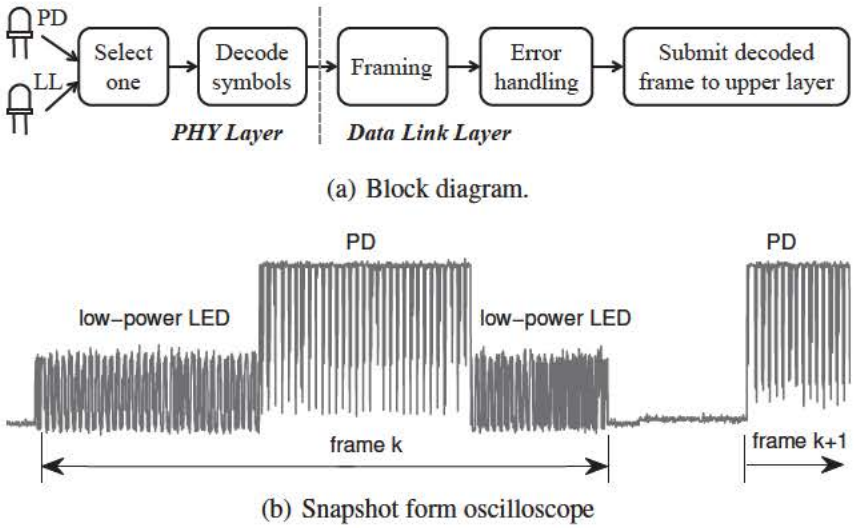
Figure 4.16: Evaluation of the proposed system in a scaled down smart car application.

that it may fail to detect the preamble and thus lose the whole frame. This is worth to be explored in the future.

### 4.6.2. Limitation and future enhancement of OpenVLC

So far, OpenVLC can only achieve a throughput of around 12 Kbp/s. The bottleneck of the low achievable throughput is due to the speed at which the symbols are sampled from the ADC by our driver (which is implemented in the Linux kernel). The Linux kernel fails to provide accurate timing once past a certain sampling speed. To solve this, adding a FPGA or micro-controller, use PSOC[3] or the programmable real-time unit of the BBB to implement the PHY layer (especially the sampling) can help.

[3]http://www.cypress.com/products/psoc-4




(a) Block diagram.

(b) Snapshot form oscilloscope

Figure 4.17: The system can decide which receiver antenna to use (switch the RX between the LL and PD) at anytime

Other limitations of the OpenVLC are the short range and narrow coverage of LL. The short range is due to the fact that this work only uses a 2.5W HL as transmitter and the transmission power is limited. This can be improved greatly by adopting common off-the-shelf brighter HLs that have higher transmission power by default. To solve the narrow coverage of LL, multiple LLs can be used to expand the reception coverage of LLs as the authors in [68] adopt to enlarge the transmission coverage.

### 4.6.3. Methods for solving PD's saturation

PD's saturation can be changed through many ways. While this work explored the impact of gain control in the transimpedance amplifier, other methods could also provide benefits, e.g., changing the reverse bias voltage applied to a PD. Dynamically changing the reverse bias voltage according to the ambient light noise and the aimed throughput is a tradeoff and worth exploring in the future. This work chooses to explore the gain control to solve PD's saturation problem because it does not impact the transmission rate.

### 4.6.4. Context information and localization

In radio systems, link quality assessment is almost solely dependent on the RSS. Upon receiving a packet, if the signal strength is low, the receiver knows that the link is weak. In VLC, it will be important not only to obtain information about the RSS of incoming packet but also about the type of antenna element used in the receiver because it can provide localization. For example, receiving information from an LED with a given RSS can be easily mapped to a location because of the directionality and the smother way in which visible lights signals decay in space compared

Table 4.3: Context information by using PD+LL as RXs (Yes/No: means the PD or LL alone can/cannot decode data successfully )

| PD | LL | Context and relative location information |
|---|---|---|
| No | No | TX and RX are not within the communication range |
| No | Yes | Noise floor around the RX is too high, link may rapidly change RSS |
| Yes | No | TX and RX are not aligned well to each other |
| Yes | Yes | TX and RX are in range of each other & noise floor is low |

to radio. Based on these information, the Data Link Layer could provide relative locations about the TX and RX and the context information of ambient environment around the RX without extra overhead. The illustration of achieved content information and relative localization is provided in Table 4.3.

## 4.7. Summary

Motivated by the vision of the Internet of Lights, which would enable LED-based devices to form their own distributed networks, this chapter has investigated the challenges associated with visible light links. It has identified the limitations and tradeoffs of two optical receivers, photodiodes and LEDs (working as receivers in reverse bias), and shown that they have complementary capabilities for networking. Based on this insight, this work has designed and implemented a new optical Data Link Layer that is resilient to dynamics caused by changes in illumination and directionality. It has evaluated the platform under key experimental setups with different dynamics, provided evidence that visible light links outperform methods relying solely on either a photodiode or an LED, and showcased the adaptability of our system in a scaled down mobile application. Our platform has an open source (driver) code and can be used by the research community to investigate the new networking challenges arising from enabling communication on LED lights.

# Part II : D2D Communication in Small Cells

# Chapter 5

# BS-Transparent D2D Communication

## 5.1. Introduction

Opportunistic scheduling [22, 23] was proposed for multiuser wireless communication networks to exploit fluctuating channel conditions, aiming to improve performance. In cellular networks, opportunistic schedulers use knowledge of the channels between Base Station (BS) and users to schedule those with favorable channel states, thus improving overall throughput.

The performance of opportunistic scheduling algorithms has been commonly investigated under the assumption of a static user population with infinitely backlogged queues [23], i.e., the BS always has data to transmit to each user. However, a more realistic setting is one with a time-varying user population and stochastic traffic loads. In such a setting, the performance of opportunistic scheduling algorithms can be very different [75, 76]. The impact of a time-varying user population is small in large cells since the BS may always have a large number of users to choose from for scheduling purposes. However, as cell sizes in future wireless networks shrink in response to increasing demands for capacity [24–26], the average number of users served by a BS will decrease and the burstiness will increase. Since opportunistic gain scales as a concave function of the user population [27], presently used scheduling algorithms are prone to losing effectiveness in small cells with dynamic traffic load.

This chapter proposes an alternate user-initiated *BS-transparent* (i.e., without changes at the BS) algorithm. It focuses on the downlink case which accounts for most of the traffic in a cellular network [77], and on http live streaming or best-effort traffic (e.g., web browsing), where mobile devices request files (chunks of content) which are then sent to users after some fetching and queueing delay. To spread traffic, this algorithm leverage the multiple radio interfaces (e.g. 3G, WiFi) available in most smartphones. The algorithm keeps tracking each user' backlogs at the BS and balances traffic requests across users, aiming to maximize the BS's long-term scheduling options and hence improve the delay performance. This improved performance on delay implicitly reduces the power consumption of cellular transmission [78].

The proposed algorithm includes a dispatcher that resides on each mobile device. To illustrate

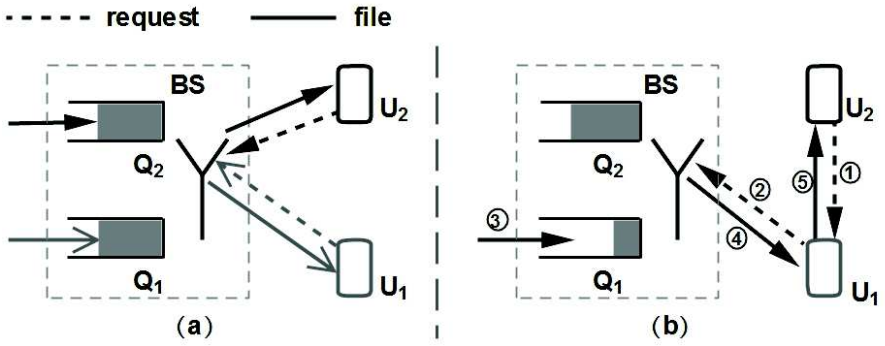



Figure 5.1: An example of traffic spreading: (a) No traffic spreading; (b) Traffic spreading from $U_2$ to $U_1$.

the traffic spreading, consider an example shown in Figure 5.1. It depicts a scenario with two users, $U_1$ and $U_2$ being served by the BS. The queues, $Q_1$ and $Q_2$, depict the number of files at users' BS queues. In this scenario, the users perceive similar channel statistics and consider below the case where they generate similar traffic loads to illustrate the spreading mechanism. In Figure 5.1(a), since the queues at the BS are balanced, the dispatchers of the users would ideally detect that traffic spreading is not beneficial. Thus users send their new requests to the BS directly. In Figure 5.1(b), there are many more files in $Q_2$ than in $Q_1$ which is nearly empty. Under this case, the dispatcher of $U_2$ would ideally detect that traffic spreading is beneficial since in the near term the risk is high that the BS has no files to send to $U_1$ and thus losses opportunistic gain. Thus, when a new request is generated by $U_2$, it forwards the request to $U_1$, who will send it to the BS. When $U_1$ receives the corresponding file from the BS, it forwards the file to $U_2$ through the user-to-user link.

The decision made by dispatchers is based on the channel statistics of all the users and their backlogs at the BS. Note that dispatchers do not exploit current channel conditions, and can not predict when a new request will be served or the instantaneous channel conditions at that time. Each user keeps track of the number of files at the BS, and shares this information periodically with other users. Users also measure their perceived channel statistics and exchange them with other users. Moreover, users are aware of, or can easily infer the BS scheduling policy and can track the destination of files received from the BS. Note that our proposed algorithm also applies to the scenarios where some users do not have their own file requests. These users will act as pure relaying nodes.

The proposed algorithm incurs additional power expenditure due to forwarding requests and files among users. Mobile devices with scarce energy resources necessitate careful power management because excessive traffic spreading can result in unacceptably high penalties in terms of power expenditure. Thus, the degree of spreading has to be carefully chosen, and the tradeoff between performance improvement and additional power expenditure must be taken into account. This chapter develops an energy-aware traffic spreading policy that can optimize the degrees of

spreading based on desired performance-energy tradeoff. The main contributions are summarized as follows:

1. Proposal of a novel traffic spreading policy to increase opportunistic gains by energy-aware user cooperation.

2. Formulation of the problem of determining the optimal spreading policy under a specified tradeoff between performance and energy as a Markov decision problem, and study properties of the corresponding optimal policy in a two-user scenario.

3. Proposal of a heuristic algorithm to reduce the computational complexity in large systems by aggregating users and using the two-user solution as a building block.

4. Based on realistic Rayleigh fading channels, simulation results are provided demonstrating that under our proposed policy: $i$) average file transfer delays be reduced by up to $20\%$ even in homogeneous scenarios using the proposed methodology; $ii$) significant gains (up to $78\%$) are typically achieved at only $20\%$ of the power expenditure of the performance-centric case; $iii$) the delay performance can be improve greatly (by up to $50\%$) in scenarios where the overall system performance is poor when some users locate far from the BS.

5. Extension of the proposed traffic spreading algorithm to large cells. Evaluation results show that in a large cell, the delay performance can be improved by up to $50\%$, and up to $73\%$ of the gain can be achieved at only $18\%$ of the maximal additional power expenditure.

The rest of this paper is organized as follows: related work is summarized in Sec. 5.2, followed by system model and dynamic programming formulation in Sec. 5.3 and 5.4 respectively. The properties of the optimal traffic spreading policy are described in Sec. 5.5, and a tractable heuristic for large systems is developed in Sec. 5.6. Simulation results and evaluation of the heuristic are presented in Sec. 5.7. Finally, the conclusions and future work are presented in Sec. 5.9.

## 5.2. Related Work

Many opportunistic scheduling algorithms taking into account users' backlogs during BS scheduling have been proposed for systems with dynamic traffic. Authors in [79–81] propose BS schedulers that try to maximize opportunistic gain as well as balance users' backlogs. Among these, [79, 80] propose the throughput-optimal MaxWeight and Exponential rules, respectively, and [81] proposes a policy named log rule to improve delay performance. All these policies react to imbalance in users' queues, sacrificing opportunistic gain in order to balance the queues. They also necessitate changes at the BS. In contrast, the policy proposed in our work is able to balance users' backlogs without sacrificing opportunistic gain, by opportunistically exploiting the BS-user and user-user channels. Moreover, our policy is *BS-transparent* and thus does not require any changes at the BS.

Some approaches are proposed to exploit both the BS-user and user-user channels, e.g., *opportunistic relaying* [82–84] and *device-to-device communication* [85–87]. Among these, [82] proposes the idea of opportunistic relaying as well as an approach that chooses the best relay maximizing the minimal quality of BS-relay and relay-user channels. In [83, 84], mobile users themselves are used as relays, instead of particular relay nodes. In [85–87], users are divided into clusters and in each cluster, a cluster header is responsible to communicate with the BS and forwards traffic to other users. Compared to ours, [82, 83, 85–87] assume users have infinitely backlogged queues. While [84] considers stochastic traffic loads as ours, none of [82–87] have investigated the delay-energy tradeoff.

As described in Sec. 5.3, this work formulates the problem of determining the optimal traffic spreading policy as a dispatching problem. Here discusses some related work on this topic. A *dispatching system* typically consists of a dispatcher and several servers. The role of the dispatcher is to route new jobs to a server based on dispatching policies. The dispatching problem has received a lot of attention since the landmark work in [88]. The author considers a homogeneous model with Poisson arrivals and exponentially distributed job size, and show that when the queue lengths of the servers (number of jobs) are known, Join the Shortest Queue (JSQ) minimizes the average waiting time in the queues. When queue lengths are unavailable, [89] shows that Round Robin is optimal. In contrast to above papers, our model only has one shared server whose service rate is affected by the dispatching policy. The model in [90] includes the case of a shared server and is the closest to ours. However, this paper like the others makes the assumption that the service rate is constant and does not depend on the instantaneous queue states. In our work, the service rate depends on the channel states as well as queue states, making the problem more complex. The emphasis in all the above papers is on performance, whereas in our case this work additionally considers the implications of the dispatching decisions on energy cost.

## 5.3. System Model

This work models the system in continuous time with $N$ users attached to a single BS, where the set of users is denoted by $\mathcal{I} = \{1, 2, ..., N\}$. The arrival requests of users are modeled as Poisson processes with mean arrival rate vector $\boldsymbol{\lambda} = \{\lambda_1, \lambda_2, ..., \lambda_N\}$, and are assumed to be independent across users. The requested file sizes of users are exponentially distributed, with mean file size vector $\boldsymbol{\theta} = \{\theta_1, \theta_2, ..., \theta_N\}$.

**Channel model:** The wireless channel is assumed to be time-varying and the channel instance between the BS and users can take values from the set $\mathcal{S} = \{s_1, s_2, \cdots, s_K\}$. Denote by $\boldsymbol{C}(t) = \{C_i(t), i \in \mathcal{I}, C_i(t) \in \mathcal{S}\}$, the vector of users' current channel states at time $t$ and by $\boldsymbol{\mathcal{C}}$, the set of all the possible channel state vectors. Further, assume the channels perceived by different users are independent. The probability that user $i$ perceives channel $s_k$ at any time is denoted by $p_i^{s_k}, s_k \in \mathcal{S}$, t. Each user $i$ shares its channel probability vector $\boldsymbol{p}_i = \{p_i^{s_1}, p_i^{s_2}, ..., p_i^{s_K}\}$ with other users. Assume all users are within the transmission range of each other, which is expected to

be the case in picocell/femtocell [91**?** ] scenarios (*the system model is also extended to large cells in Sec. 5.8, where not all users can communicate with each other*). Define a non-negative value $R_i^{s_k}$ for each channel state $s_k \in \mathcal{S}$, which denotes the data rate (rate supported by the channel) in bits/second of user $i$.

Our system model consists of three main components, i.e., the BS scheduler, the queues at the BS, and a dispatcher that models the joint behavior of all the mobile devices. The BS maintains a separate queue corresponding to each user, and the number of files waiting to be sent by the BS to each user at time $t$, i.e., the number of pending requested files, is denoted by $\boldsymbol{Q}(t) \equiv (Q_i(t), i \in \mathcal{I}) \in \mathbb{Z}_+^N$, . This work follows the convention that random variables are denoted by capital letters (i.e., $\boldsymbol{Q}(\cdot)$ and $\boldsymbol{C}(\cdot)$), while the possible values are denoted by the corresponding small letters (i.e., $\boldsymbol{q}$ and $\boldsymbol{c}$).

**Scheduling policy:** The BS scheduling policy is modeled through $\xi_i(\boldsymbol{q}, \boldsymbol{c})$, denoting the probability that user $i$ is selected to be served by the scheduler, conditional on the queues being in state $\boldsymbol{q}$, and channel vector being $\boldsymbol{c}$. The average queue state-dependent service rate of user $i$ is denoted as $\mu_i(\boldsymbol{q})$. Two channel-aware scheduling policies are considered: a queue-unaware policy where $\xi_i(\boldsymbol{q}, \boldsymbol{c})$ only depends on the set of non-zero elements in $\boldsymbol{q}$, and a queue-aware policy that have a stronger dependence on $\boldsymbol{q}$:

#### 5.3.0.1. Queue-unaware, greedy scheduling policy

Queue-unaware means the scheduler is unaware of the queue length, but knows whether a queue is empty or not. At any time $t$, a greedy scheduler chooses a non-empty queue $i$ to serve if user $i$ has the largest instantaneous data rate.

#### 5.3.0.2. Queue-aware, log rule scheduling policy [81]

Queue-aware means the scheduler is aware of the queue length. At time $t$, a log rule scheduler makes decisions based on current channel state and the logarithm of queue length, i.e., choosing user $i$ that satisfies

$$i \in \arg\min_{j \in \mathcal{I}} \frac{R_j^{c_j(t)}}{\sum_{k \in \mathcal{S}} p_j^{s_k} R_j^{s_k}} \log(b + a_j Q_j(t)) \ ,$$

where $b$ and $a_j$ are constants.

**Dispatching policy:** The dispatching policy used across all users when the local communication among users is not congested is defined through the probability matrix $\boldsymbol{\sigma}(\boldsymbol{q})$ as:

$$\boldsymbol{\sigma}(\boldsymbol{q}) = \begin{bmatrix} \sigma_1^1(\boldsymbol{q}) & \cdots & \sigma_1^N(\boldsymbol{q}) \\ \cdots & \cdots & \cdots \\ \sigma_N^1(\boldsymbol{q}) & \cdots & \sigma_N^N(\boldsymbol{q}) \end{bmatrix}, \tag{5.1}$$

where $\sigma_j^i(\boldsymbol{q})$ denotes the probability of dispatching user $j$'s request to user $i$, conditional on the queues being in state $\boldsymbol{q}$. If the local communication is congested, then users will never dispatch

their requests to other users. The set of all the possible dispatching policies is defined as

$$\mathcal{A} \equiv \{\boldsymbol{\sigma}(\boldsymbol{q}) : \sum_{i \in \mathcal{I}} \sigma_j^i(\boldsymbol{q}) = 1,\ 0 \leq \sigma_j^i(\boldsymbol{q}) \leq 1,\ j \in \mathcal{I}\}\ . \tag{5.2}$$

The rate at which files arrive to user $i$'s queue at the BS is denoted by $\lambda_i'(\boldsymbol{q}, \boldsymbol{\sigma}(\boldsymbol{q}))$. This rate corresponds to the rate of requests sent by user $i$ to the BS (including forwarded requests from other users), as shown in Figure 5.2 (a). The insight of the dispatching policy is illustrated in Figure 5.2 (b), where it depicts a two-user scenario and the axes represent users' BS queues. The $\mu_i'$ and $\mu_i$ are user $i$'s service rate when its queue is and is not empty, respectively. It is straightforward that $\mu_i' \geq \mu_i, i \in \mathcal{I}$. As opportunistic gain scales with the number of users [27], then $\sum_{i=1}^{2} \mu_i \geq \mu_j', \forall j \in \{1, 2\}$, namely, opportunistic gain decreases when queue state reaches the axes. Our proposed dispatching policy dynamically controls the direction of vector $\boldsymbol{\lambda}'$ to keep as many queues being non-empty as possible, which helps scheduling policies to increase the opportunistic gain.

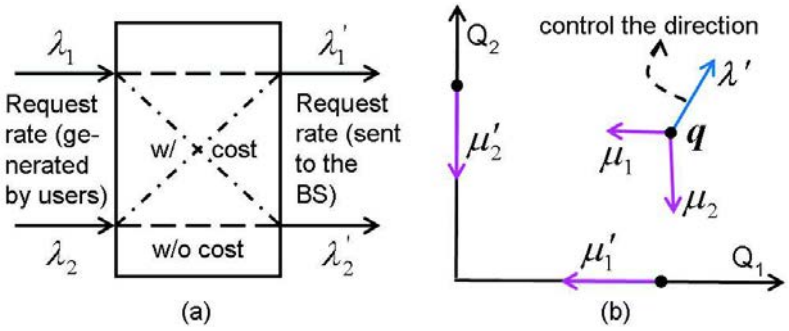



Figure 5.2: Proposed dispatching policy: (a) the dispatcher; (b) the dispatching algorithm controls the arrivals to users' BS queues.

The proposed policy also applies to multicast. The difference in multicast is all the users subscribing the same multicast channel have identical arrival requests. Instead of sending the same request by all users, they can choose the one with the shortest BS queue to send a file request. This chosen user then forwards the corresponding file to others after receiving it.

**Performance metrics:** The used metrics are average file transfer delay and *the re-routing cost*, i.e., additional power expenditure induced by traffic spreading. Assume there is no queueing delay for the user-to-user transfer, but each transfer certainly adds additional forwarding delay and power expenditure. The additional delay and power expenditure incurred for a single file of user $j$ is modeled as a function of its mean file size $\theta_j$, i.e., $\eta_j^i(\theta_j)$ and $\phi_j^i(\theta_j)$, and define $\eta_j^j(\theta_j) = 0$, $\phi_j^j(\theta_j) = 0$. Note that a very high $\phi_j^i(\theta_j)$ can be used to model the case where the large distance between users makes the communication between them infeasible. Moreover, the average additional forwarding delay and power expenditure incurred for re-routing files of user $j$ are denoted by $\bar{\eta}_j^i$ and $\bar{\phi}_j^i$, $i, j \in \mathcal{I}$, respectively. The objective function this work seeks to

minimize is

$$\bar{D} + \sum_{j\in\mathcal{I}}\sum_{i\in\mathcal{I}} \bar{\eta}_j^i + \sum_{j\in\mathcal{I}}\sum_{i\in\mathcal{I}} w_j^i \cdot \bar{\phi}_j^i \,, \tag{5.3}$$

where $\bar{D}$ is the average BS-user delay and $\boldsymbol{w} = \{w_j^i, i, j \in \mathcal{I}\}$ is the $N \times N$ weight matrix associated with the power expenditure of users that determines the tradeoff between delay and power expenditure.

## 5.4. Dynamic Programming Formulation

Consider the process $(\boldsymbol{Q}(t), t \geq 0)$ initiated in state $\boldsymbol{Q}(0)$ and evolving under a dispatching policy $\boldsymbol{\sigma}$ and a scheduling policy $\boldsymbol{\xi}$. Define the vector $\boldsymbol{\mu(q)} \equiv (\mu_i(\boldsymbol{q}), i \in \mathcal{I})$ as the average service rate of users, if the queue state is $\boldsymbol{q}$. Clearly, $\boldsymbol{\mu(q)}$ depends on the scheduling policy used at the BS. For a given scheduling policy, the average service rate $\mu_i(\boldsymbol{q})$ is given as

$$\mu_i(\boldsymbol{q}) = \sum_{\boldsymbol{c}\in\boldsymbol{\mathcal{A}}} \left( \prod_{j\in\mathcal{I}} p_j^{c_j} \right) \xi_i(\boldsymbol{q}, \boldsymbol{c}) \cdot R_i^{c_i} \,. \tag{5.4}$$

Assume over an epoch, each queue $i \in \mathcal{I}$ is served at constant service rate $\mu_i(\boldsymbol{q})$. Since the channel varies much faster than the queue dynamics,the service rate averaged across channel fluctuations at each queue state is used. A rigorous justification of the service rate with consideration of packet or file dynamics can be found in [92]. If the process is in state $\boldsymbol{q}$ and under the dispatching policy $\boldsymbol{\sigma}(\boldsymbol{q})$, the file arrival rate to $Q_i$ is given by

$$\lambda_i'(\boldsymbol{q}, \boldsymbol{\sigma}(\boldsymbol{q})) = \sum_{j\in\mathcal{I}} \sigma_j^i(\boldsymbol{q})\lambda_j, \quad i \in \mathcal{I} \,. \tag{5.5}$$

Our objective is to find the right $\boldsymbol{\sigma}(\boldsymbol{q}) \in \mathcal{A}$ for each state $\boldsymbol{q}$ that minimizes (5.3). Using Little's law, (5.3) becomes

$$\frac{\mid E\left[\boldsymbol{Q}\right] \mid}{\mid \boldsymbol{\lambda} \mid} + \sum_{j\in\mathcal{I}}\sum_{i\in\mathcal{I}} \bar{\eta}_j^i + \sum_{j\in\mathcal{I}}\sum_{i\in\mathcal{I}} w_j^i \cdot \bar{\phi}_j^i \,, \tag{5.6}$$

where $|\cdot|$ denotes the $L_1$ norm.

Under a fixed policy $\boldsymbol{\sigma}(\boldsymbol{q})$, the process $(\boldsymbol{Q}(t), t \geq 0)$ is a Markov process on $\mathbb{Z}_+^N$ with state-dependent (depends on both channel and queue states) transition rate. For convenience, $\boldsymbol{Q}(t)$ is uniformed following [93]. For any $\boldsymbol{q} \in \mathbb{Z}_+^N$, make the following definitions:

$$D_i\boldsymbol{q} \equiv \max(\boldsymbol{0}, \boldsymbol{q} - \boldsymbol{e}_i) \,, \qquad A_i\boldsymbol{q} \equiv \boldsymbol{q} + \boldsymbol{e}_i \,, \tag{5.7}$$

where $\boldsymbol{e}_i$ is a $1 \times N$ zero-valued vector except the $i^{th}$ element is 1 and max is the element-wise maximum operation. The $D_i$ in (5.7) denotes a file is successfully transmitted from the BS to user

$i$, and $A_i$ means a file arrives to $Q_i$ at the BS.

Let $\varphi \geq |\boldsymbol{\lambda}| + \max_{\boldsymbol{q}} |\boldsymbol{\mu}(\boldsymbol{q})|$. Let $\tau_k$ denote the time of the $k^{th}$ transition of $\boldsymbol{Q}(t)$ and $\tau_0 = 0$. Also, let $\boldsymbol{Q}_k = \lim_{t \downarrow \tau_k} \boldsymbol{Q}(t)$. Then, under policy $\boldsymbol{\sigma}(\boldsymbol{q})$, the process $\boldsymbol{Q}(t)$ can be viewed as having a state-independent event transition rate of $\varphi$, and the transition probabilities are given by

$$P\big(\boldsymbol{Q}_{k+1} = A_i\boldsymbol{q} \mid \boldsymbol{Q}_k = \boldsymbol{q}\big) = \lambda_i'(\boldsymbol{q}, \boldsymbol{\sigma}(\boldsymbol{q}))/\varphi \,, \tag{5.8}$$

$$P\big(\boldsymbol{Q}_{k+1} = D_i\boldsymbol{q} \mid \boldsymbol{Q}_k = \boldsymbol{q}\big) = \mu_i(\boldsymbol{q})/\varphi \,, \tag{5.9}$$

$$P\big(\boldsymbol{Q}_{k+1} = \boldsymbol{q} \mid \boldsymbol{Q}_k = \boldsymbol{q}\big) = 1 - \big(|\boldsymbol{\lambda}| + |\boldsymbol{\mu}(\boldsymbol{q})|\big)/\varphi. \tag{5.10}$$

Moreover, define the function $\boldsymbol{f}(\cdot)$ as

$$\boldsymbol{f}\left(\boldsymbol{q}, \boldsymbol{\sigma}(\boldsymbol{q})\right) = \sum_{j \in \mathcal{I}} \sum_{i \in \mathcal{I}} \sigma_j^i(\boldsymbol{q}) \cdot \left[\eta_j^i(\theta_j) + w_j^i \cdot \phi_j^i(\theta_j)\right] \,. \tag{5.11}$$

The cost (our objective in (5.6)) under policy $\boldsymbol{\sigma}$ over $[0, \tau_k)$ when starting from an initial queue state $\boldsymbol{q}$ is

$$\mathbb{E}_{\boldsymbol{q}}^{\boldsymbol{\sigma}}\left[\int_0^{\tau_k} \left[\frac{|\,\boldsymbol{Q}(t)\,|}{|\,\boldsymbol{\lambda}\,|} + \boldsymbol{f}\big(\boldsymbol{Q}(t), \boldsymbol{\sigma}(\boldsymbol{Q}(t))\big)\right] dt\right], \tag{5.12}$$

which, by ignoring the constant multiplier $\varphi^{-1}$, is equal to:

$$V_k^{\boldsymbol{\sigma}}(\boldsymbol{q}) \equiv \mathbb{E}_{\boldsymbol{q}}^{\boldsymbol{\sigma}}\left[\sum_{l=0}^{k-1} \left(\frac{|\,\boldsymbol{Q}_l\,|}{|\,\boldsymbol{\lambda}\,|} + \boldsymbol{f}\left(\boldsymbol{Q}_l, \boldsymbol{\sigma}(\boldsymbol{Q}_{\boldsymbol{l}})\right)\right)\right] \,. \tag{5.13}$$

Then the average cost under policy $\boldsymbol{\sigma}$ when starting from state $\boldsymbol{q}$ is given as follows:

$$J_{\boldsymbol{q}}^{\boldsymbol{\sigma}} = \lim_{k \to \infty} \sup \frac{1}{k} V_k^{\boldsymbol{\sigma}}(\boldsymbol{q}) \,. \tag{5.14}$$

The objective function given in (5.6) seeks to find the minimal average cost and the corresponding optimal control, which fits the classical dynamic programming (refer to Sec. 7.4 of [93]). Under all possible dispatching probabilities $\boldsymbol{\sigma}(\boldsymbol{q}) \in \mathcal{A}$, the minimal average cost $J^*$ is well-defined, independent of the initial state $\boldsymbol{Q}(0)$, and satisfies Bellman's equation:

$$\begin{aligned} J^* = \min_{\boldsymbol{\sigma}(\boldsymbol{q}) \in \mathcal{A}} \Bigg\{ & \frac{|\,\boldsymbol{q}\,|}{|\,\boldsymbol{\lambda}\,|} + \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{\sigma}(\boldsymbol{q})) \\ & + \mathbb{E}^{\boldsymbol{\sigma}(\boldsymbol{q})}\big\{\big[h\big(\boldsymbol{Q}_{k+1}\big) - h\big(\boldsymbol{Q}_k\big)\big] \mid \boldsymbol{Q}_k = \boldsymbol{q}\big\}\Bigg\} \\ = & \frac{|\,\boldsymbol{q}\,|}{|\,\boldsymbol{\lambda}\,|} + \sum_{i \in \mathcal{I}} \frac{\mu_i(\boldsymbol{q})}{\varphi}\Big[h\big(D_i\boldsymbol{q}\big) - h\big(\boldsymbol{q}\big)\Big] + \min_{\boldsymbol{\sigma}(\boldsymbol{q}) \in \mathcal{A}} \sum_{j \in \mathcal{I}} \sum_{i \in \mathcal{I}} \\ & \frac{\lambda_j \sigma_j^i(\boldsymbol{q})}{\varphi}\Big\{\eta_j^i(\theta_j) + w_j^i \cdot \phi_j^i(\theta_j) + \big[h\big(A_i\boldsymbol{q}\big) - h\big(\boldsymbol{q}\big)\big]\Big\} \,, \end{aligned} \tag{5.15}$$

where $h(\boldsymbol{q}) = J(\boldsymbol{q}) - J(\boldsymbol{q}_s)$ is a relative cost function with $\boldsymbol{q}_s$ being a reference state. The

optimal dispatching policy $\boldsymbol{\sigma}^*(\boldsymbol{q})$ that minimizes (5.15) can be calculated through methods such as the value iteration or policy iteration from the dynamic programming framework [93].

## 5.5. Properties of the Optimal Policy

This section presents some properties of the optimal dispatching policy $\boldsymbol{\sigma}^*(\boldsymbol{q})$when the local communication among users is not congested. The cellular link has a Rayleigh fading channel and the Signal-to-Noise-Ratio (SNR) is assumed to be constant during a time slot. Denote the channel bandwidth as $B$, and the distance between user $i$ and the BS as $d_i$, both of which affect users' data rate. For the channel being in state $s_k \in \mathcal{S}$, the data rate $R_i^{s_k}$ is given by the Shannon formula with a 3dB SNR loss (to model achievable data rate):

$$R_i^{s_k} = B \cdot \log_2(1 + \text{SNR}_i(s_k)/2) \; . \tag{5.16}$$

The settings of channel parameters, mean file size and additional power expenditure for spreading files are presented in Sec. 5.7. Note that the properties of the optimal policy are not sensitive to these settings.

### 5.5.1. Restricting the Optimal Policy Space

The following theorem guarantees that the optimal value of the objective function can be achieved by non-randomized policies that apply deterministic rules for dispatching the arrivals at a given system state. This reduces the computational effort required to compute an optimal dispatching policy, and allows us to use value iteration in the sequel to study the structure of the optimal policy.

**Theorem 1.** *There exists an optimal dispatching policy* $\boldsymbol{\sigma}^*(\boldsymbol{q})$ *such that each element* ${\sigma^*}_j^i \in \{0, 1\}$.

*Proof*: From Sec. 5.4, it has already been know that a dispatching policy that minimizes (5.15) is an optimal policy. To minimize (5.15), it is sufficient to minimize the last "min" part. Since the dispatching rule used by each user is chosen independent of the others, then only each part within the first sum has to be minimized, namely

$$\min_{\boldsymbol{\sigma}(\boldsymbol{q}) \in \mathcal{A}} \sum_{i \in \mathcal{I}} \sigma_j^i(\boldsymbol{q}) \Big\{ \eta_j^i(\theta_j) + w_j^i \cdot \phi_j^i(\theta_j) + \big[h\big(A_i \boldsymbol{q}\big) - h\big(\boldsymbol{q}\big)\big] \Big\},$$

where $i, j \in \mathcal{I}$. Below consider a particular value of $j$. Under queue state $\boldsymbol{q}$, denote:

$$\alpha_i \equiv \sigma_j^i(\boldsymbol{q}), \quad \beta_i \equiv \eta_j^i(\theta_j) + w_j^i \cdot \phi_j^i(\theta_j) + \big[h\big(A_i \boldsymbol{q}\big) - h\big(\boldsymbol{q}\big)\big]$$

Furthermore, let $\boldsymbol{\alpha} = \{\alpha_i, i \in \mathcal{I}\}$ denote a stochastic vector. To prove the theorem, it needs to show that for a given $\boldsymbol{\beta}$, the minimal value of $\boldsymbol{\alpha} \cdot \boldsymbol{\beta}$ can be achieved when $\alpha_{i^*} = 1$, where

$i^* \in \arg\min_{i\in\mathcal{I}}\{\beta_i\}$, augmented with a tie-breaking rule. For $i^*$ and $\forall i \in \mathcal{I}$, then

$$1 \cdot \beta_{i^*} = \alpha_{i^*}\beta_{i^*} + \sum_{i\neq i^*} \alpha_i\beta_{i^*} \leq \alpha_{i^*}\beta_{i^*} + \sum_{i\neq i^*} \alpha_i\beta_i \tag{5.17}$$

Therefore when the element $\alpha_{i^*}$ of vector $\boldsymbol{\alpha}$ is set to 1 and all the other elements of $\boldsymbol{\alpha}$ are set to 0, the minimal value of $\boldsymbol{\alpha}\cdot\boldsymbol{\beta}$ is achieved, which proves the theorem. ■

### 5.5.2. A Two-user System

From Theorem 1, it can be known that under the two-user model and at any queue state $\boldsymbol{q}$, there are three reasonable controls: *i)* $\sigma(\boldsymbol{q}) = [1\,0; 0\,1]$; *ii)* $\sigma(\boldsymbol{q}) = [1\,0; 1\,0]$; *iii)* $\sigma(\boldsymbol{q}) = [0\,1; 0\,1]$. In the rest of this paper, these controls are referred as *no re-routing*, $U_2 \rightarrow U_1$ and $U_1 \rightarrow U_2$, respectively. The optimal dispatching policies evaluated numerically under different scenarios are shown in Figure 5.3. The axes correspond to the number of files in the users' queues, and the figure depicts the optimal dispatching strategy at each state. From this figure, the following properties can be observed:

**Existence of switching curves:** As Figure 5.3 shows, the optimal policy in all the above cases consists of a set of switching curves, i.e., the policy is transition monotone [94]. Here, switching curves refer to the boundaries between contiguous regions where the same control is used in each state of the region. It can be conjectured that an optimum policy can be described by threshold values $q_2^a$ and $q_2^b$ corresponding to each value of $q_1$, such that

$$\sigma^*(q_1, q_2) = \begin{cases} U_1 \rightarrow U_2, & \text{if } q_2 \leq q_2^a \\ \textit{No re-routing}, & \text{if } q_2^a \leq q_2 \leq q_2^b \\ U_2 \rightarrow U_1, & \text{if } q_2 \geq q_2^b \end{cases}$$

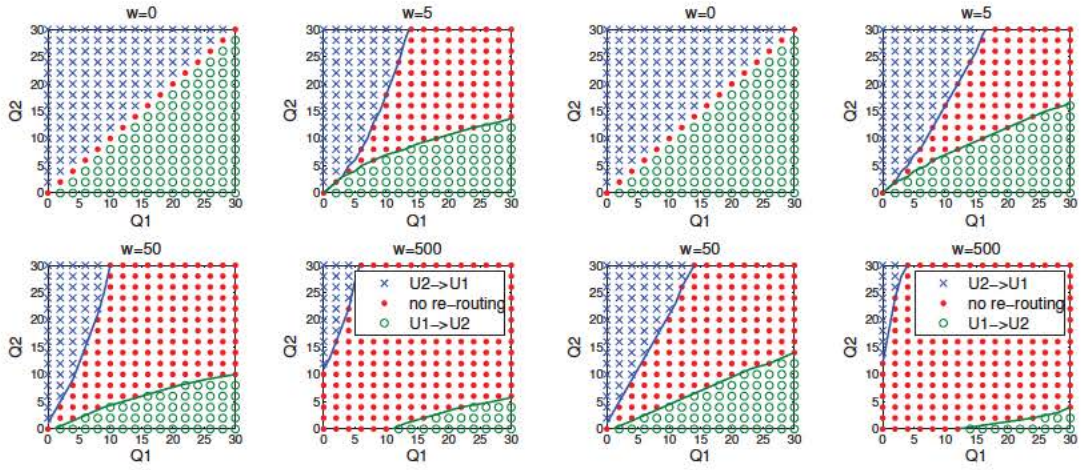
For the two-user homogeneous scenario under a two-state channel model with delay-optimal scheduling policy, the optimal policy can be proved that it indeed possesses this structure, as shown in the following theorem:

**Theorem 2.** *There exists an optimal dispatching policy that has switching curves (transition monotone), under the two-user homogeneous scenarios under a two-state channel model (on/off) with delay-optimal scheduling policy.*

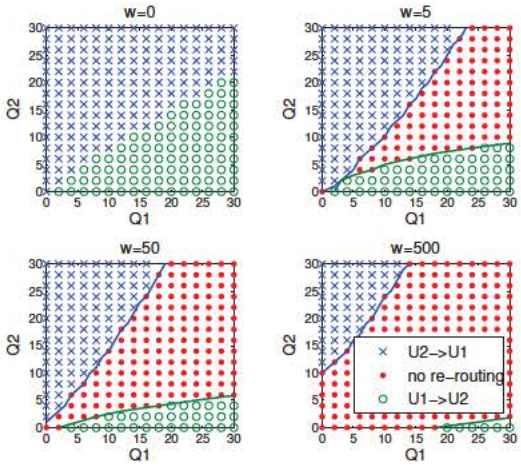
Introduction of some notations and the value iteration method. For the two-state channel (*on/off*), the probability that user $i$'s channel is *on* was denoted as $p_i$, $i \in \{1, 2\}$. Since the homogeneous scenario is considered, then

$$\eta \cdot z_j^i = \eta_j^i(\theta_j), \quad \phi \cdot z_j^i = \phi_j^i(\theta_j), \quad \forall i, j \in \{1, 2\},$$

where $z_j^i = 1$ if $j \neq i$ and $z_j^i = 0$ if $j = i$.




(a) $\lambda = 0.2$ arrivals/sec; (b) $\lambda = 0.2$ arrivals/sec;




(c) $\lambda = \{0.19, 0.21\}$ arrivals/sec;

Figure 5.3: Optimal dispatching strategy as a function of queue backlogs: (a) Homogeneous scenario ($d = 100$ m), greedy scheduler; (b) Homogeneous scenario ($d = 100$ m), the log rule scheduler; (c) Heterogeneous scenario ($d = \{92, 100\}$ m), greedy scheduler.

The value iteration is widely used to solve the Bellman equation. The most used version of the value iteration method for the average cost problem is to select an initial state and generate successively the corresponding optimal $k$-stage cost $J_k(\boldsymbol{q})$. As shown in [93], the ratios $J_k(\boldsymbol{q})/k$ converges to the optimal average cost per stage $J^*$ as $k \to \infty$. Therefore, induction upon the value iteration method can be used to prove the theorem, as the author does in [93]. First define the relative value $h_k(\boldsymbol{q})$ at stage $k$ given by

$$h_k(\boldsymbol{q}) = J_k(\boldsymbol{q}) - J_k(\boldsymbol{q}_s) \tag{5.18}$$

where $\boldsymbol{q}_s$ is a reference state. Moreover, define $\Delta_k(\boldsymbol{q})$ as follows:

$$\begin{aligned}\Delta_k(\boldsymbol{q}) =& h_k(\boldsymbol{q}+\boldsymbol{e}_1) - h_k(\boldsymbol{q}+\boldsymbol{e}_2)\\ =& J_k(\boldsymbol{q}+\boldsymbol{e}_1) - J_k(\boldsymbol{q}_s) - (J_k(\boldsymbol{q}+\boldsymbol{e}_2) - J_k(\boldsymbol{q}_s))\\ =& J_k(\boldsymbol{q}+\boldsymbol{e}_1) - J_k(\boldsymbol{q}+\boldsymbol{e}_2)\end{aligned} \tag{5.19}$$

The following lemma supports the proof of Theorem 2.

**Lemma 1.** *The $\Delta_k(\boldsymbol{q})$ is monotonically non-decreasing in $q_1$ for each fixed $q_2$, where $\boldsymbol{q} \equiv \{q_1, q_2\}, q_1, q_2 \in \mathbb{Z}_+$.*

The proof of Lemma 1 is presented in the Appendix.

*Proof*: [Proof of Theorem 2] Consider the optimal dispatching policy characterized by Theorem 1. To show the optimal policy has switching curves for the two-user homogeneous scenarios, it is sufficient to show that $\Delta_k(\boldsymbol{q})$ is monotonically non-decreasing in $q_i$ for each fixed $q_j$, $i, j \in \{1, 2\}$ and $i \neq j$ [93], which can be obtained from Lemma 1. ■

**Performance vs. Energy consumption:** Choosing a weight of $\mathbf{0}$ implies that the optimal policy is one which minimizes average file transfer delay. In the case of the homogeneous scenarios of Figure 5.3 (a) and (b), this corresponds to dispatching arrivals to the shortest queue, as described in [88]. In the heterogeneous case of Figure 5.3 (c), arrivals are dispatched rather to the queue with less backlog, taking into account the difference in average service rates. A higher value of the weight, $\boldsymbol{w}$, implies that delay performance is sacrificed in order to reduce the excess power expenditure due to traffic spreading. It can be observed that the regions corresponding to *re-routing areas* ($U_1 \rightarrow U_2$ and $U_2 \rightarrow U_1$) diminish progressively as the weight attached to power expenditure increases. At very high values of $\boldsymbol{w}$, traffic spreading is initiated only when the imbalance between the user queues is very large.

**Switching curve shape:** It can be observed from the results in Figure 5.3 that the level of imbalance between the queues that is required for arrival re-routing to be the optimal strategy increases as the overall backlog increases. For instance, the threshold on the queue length of $U_2$ beyond which arrivals are re-routed to $U_1$ appears to be a convex, increasing function of the backlog in $Q_1$. The intuition behind this is that when the backlog in both queues is large, the time interval for a queue to empty out is likely to be long, and the shorter queue might yet see many arrivals even without re-routing. In such a case, the gain from dispatching requests to other users to balance the queues does not justify the associated power expenditure.

**Dispatching as a function of the scheduling policy:** It can be observed that the optimal dispatching policy under the log rule scheduler (Figure 5.3(b)) consists of switching curves that favor more re-routing, especially at larger queue-lengths. For example, the threshold $q_2^b$ prompting re-routing is lower, and does not increase as rapidly with total queue length as under the greedy scheduler (Figure 5.3(a)). The log rule scheduler itself reacts to imbalance in queues, sacrificing opportunistic gain in order to balance the queues. The optimal dispatcher takes this into account,

resulting in the above policy. However, as will be seen in the sequel, the dispatcher complements the log rule scheduler well and the combination does achieve better performance at lower power expenditure compared to the queue-unaware greedy scheduler.

**Impact of the request arrival rate:** Switching curves of the optimal dispatching policy under the greedy scheduler for different request arrival rates are shown in Figure 5.4. It can be observed that with the increase of arrival rate, the re-routing areas decrease. One reason for this is that at lower arrival rates, the shorter queue could be emptied before the next arrival to either user. Thus, the optimal dispatcher is more aggressive at re-routing requests even when queue lengths are larger.

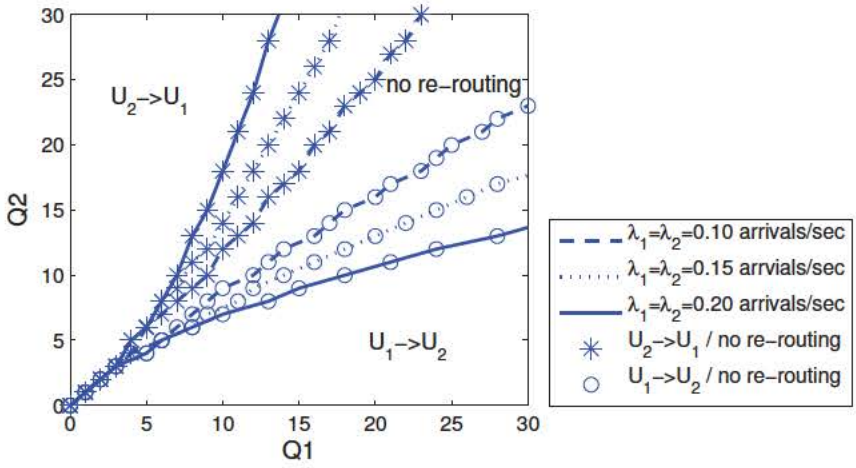



Figure 5.4: Switching curves of the optimal dispatching policy under different request arrival rates where $d = 100$ m and $w = 5$.

## 5.6. A Heuristic Algorithm for Multi-user System

In scenarios with many users, numerically computing the optimal dispatching policy by solving the dynamic programming formulation (5.15) becomes intractable. Thus a heuristic algorithm is proposed to determine the dispatching policy for a multi-user system that uses the dynamic programming solution of two-user scenarios as a building block. Here, I focus on currently deployed queue-unaware schedulers such as the greedy policy. The heuristic used to compute dispatching decisions at user $i$ is specified below in Algorithm 4. Denote by $dp(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\phi}})$, the optimal dispatching policy for the two-user model, where $\tilde{\boldsymbol{\lambda}} = \{\tilde{\lambda}_1, \tilde{\lambda}_2\}$ is the vector of arrival rates, $\tilde{\boldsymbol{\mu}}(\tilde{\boldsymbol{q}}) = \{\tilde{\mu}_1(\tilde{\boldsymbol{q}}), \tilde{\mu}_2(\tilde{\boldsymbol{q}})\}, \tilde{\boldsymbol{q}} \in \mathbb{Z}_+^2$ specifies the queue-state dependent service rates and $\tilde{\boldsymbol{\phi}} = \{\tilde{\phi}_1^2(\theta_1), \tilde{\phi}_2^1(\theta_2)\}$ is the re-routing power expenditure.

The proposed heuristic considers two options for a new request, i.e., forwarding it directly to the BS or dispatching it to the user with the least amount of work (workload) in its BS queue. To this end, all users other than the one with the least workload are treated as a single combined

**Algorithm 4** A Heuristic Algorithm for N-user System

// Heuristic to dispatch a new arrival at user $i$.
// **Definitions:** 1) $\boldsymbol{e}_l \equiv$ a $1 \times N$ zero-valued vector except the $l^{th}$ element is 1; 2) $\mathbf{1}_N \equiv \sum_{l=1}^{N} \boldsymbol{e}_l$.
**Input:** $\boldsymbol{\lambda}$ **:** a $1 \times N$ vector of arrival rates
$\boldsymbol{q}$ : a $1 \times N$ vector of queue state
$\boldsymbol{\mu}(\boldsymbol{q})$: a matrix of queue-state dependent service rates
**Output:** dispatching decision at user $i$

$j \leftarrow \arg\min_{l \in \mathcal{I}} \{q_l / \mu_l(\boldsymbol{e}_l)\}$.
**if** $j \neq i$ **then**
  $\mathcal{Y}_S \leftarrow \{j\}$; $\mathcal{Y}_B \leftarrow \mathcal{I} \setminus \mathcal{Y}_S$; $\mathcal{Y}_P \leftarrow \emptyset$
  $\tilde{q}_1 \leftarrow \sum_{l \in \mathcal{I} \setminus \mathcal{Y}_S} q_l$; $\tilde{q}_2 \leftarrow q_j$
  $\tilde{\boldsymbol{\mu}}_1(\tilde{\boldsymbol{q}}) \leftarrow \begin{cases} 0, & \tilde{q}_1 = 0 \\ \sum_{l \neq j} \mu_l(\mathbf{1}_N - \boldsymbol{e}_j), & \tilde{q}_1 > 0, \tilde{q}_2 = 0 \\ \sum_{l \neq j} \mu_l(\mathbf{1}_N), & \tilde{q}_1 > 0, \tilde{q}_2 > 0 \end{cases}$
  $\tilde{\boldsymbol{\mu}}_2(\tilde{\boldsymbol{q}}) \leftarrow \begin{cases} 0, & \tilde{q}_2 = 0 \\ \mu_j(e_j), & \tilde{q}_1 = 0, \tilde{q}_2 > 0 \\ \mu_j(\mathbf{1}_N), & \tilde{q}_1 > 0, \tilde{q}_2 > 0 \end{cases}$
  **while** $\mathcal{Y}_B \neq \emptyset$ **do**
    $i^* \leftarrow \arg\max_{l \in \mathcal{Y}_B} \{q_l / \mu_l(\boldsymbol{e}_l)\}$
    $\tilde{\lambda}_1 \leftarrow \sum_{l \in \mathcal{I} \setminus \mathcal{Y}_S} \lambda_l$; $\tilde{\lambda}_2 \leftarrow \sum_{l \in \mathcal{Y}_S} \lambda_l$
    $\tilde{\boldsymbol{\phi}} \leftarrow \{\phi^j_{i^*}(\theta_{i^*}), \phi^{i^*}_j(\theta_j)\}$
    $\boldsymbol{\sigma} \leftarrow dp(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\phi}})$
    **if** $\boldsymbol{\sigma}(\tilde{q}_1, \tilde{q}_2) = U_1 \rightarrow U_2$ **then**
      **if** $i^* = i$ **then**
        **return** dispatch the new request to user $j$
      **else**
        $\mathcal{Y}_B \leftarrow \mathcal{Y}_B \setminus i^*$; $\mathcal{Y}_S \leftarrow \mathcal{Y}_S \cup i^*$
      **end if**
    **else if** $i^* = i$ **then**
      **return** send the new request directly to the BS
    **else**
      $\mathcal{Y}_B \leftarrow \mathcal{Y}_B \setminus i^*$; $\mathcal{Y}_P \leftarrow \mathcal{Y}_P \cup i^*$
    **end if**
  **end while**
**else**
  **return** send the new request directly to the BS
**end if**

user, and their queues are also treated as a single combined queue. A series of two-user dynamic programming formulations are solved, where the two users are the combined user and the user with least workload. In order to determine the parameters of the two-user dynamic program, the states where the combined queue is non-empty are mapped to states where all the component queues are non-empty in the multi-user system. The service rate of the combined queue at a state is calculated as the sum of the service rates of the component queues in the corresponding state

(steps 4-6).

The users are examined in sequence, and the dispatching strategy is decided in order of decreasing workload. The arrival rates to the combined queue reflects the dispatching decisions made at all the users with higher workload than the one currently under consideration (step 9). The dynamic programming solution is computed taking into account the power expenditure associated with dispatching from the current user to the one with least workload. The state considered in the reduced dynamic program is always one where the combined user queue length is the sum of the queue lengths of the component queues. Arrivals to the current user are dispatched to the user with least workload in the multi-user system if the optimal policy in the reduced scenario is to re-route from the combined queue to the other. Note that the worst-case time-complexity of the above heuristic to obtain the dispatching decision for a new request is $O(N-1)$.

## 5.7. Performance Evaluation

This section evaluates the proposed traffic spreading policy through simulations and demonstrate the tradeoff between performance improvement and additional power expenditure resulting from the dynamic programming formulation and our multi-user heuristic. Assume the mean file size $\theta_j$=1MB, under which the additional power expenditure $\phi_j^i(\theta_j)$ is 1 Joule for all $i \neq j, i, j \in \mathcal{I}$. The settings of the channel parameters are listed in Table 5.1. Ignore the user-user delay (i.e., $\eta_j^i = 0, \forall i, j \in \mathcal{I}$) (thus the local communication among users is uncongested) since the data rate of user-user link would be much higher than the cellular link in reality [95]. The average file transfer delay and additional power expenditure are estimated within a relative error of 2%, at a confidence interval of 95%.

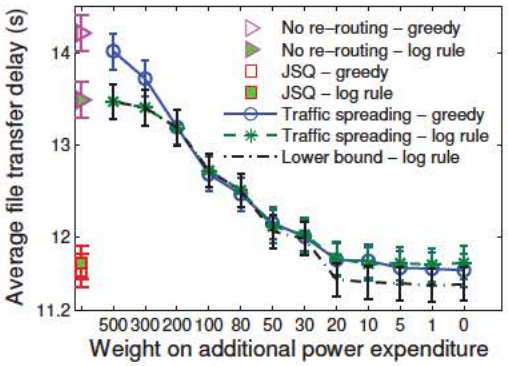
Table 5.1: CHANNEL PARAMETERS

| Parameters | Value |
|---|---|
| Bandwidth (Min bandwidth in LTE) | 1.4 MHz |
| BS Tx power spectral density | $0.1/1.4$ W/MHz |
| Noise spectral density | $10^{-8}/1.4$ W/MHz |
| Path loss exponent (Urban Area) | 3 |
| Slot time | 10 ms |
| Doppler shift (ITU Pedestrian A) | 5 Hz |

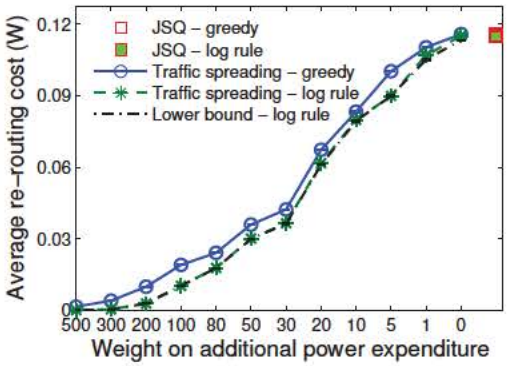
To evaluate the proposed traffic spreading, it is compared with two dispatching policies:

#### 5.7.0.1. No re-routing

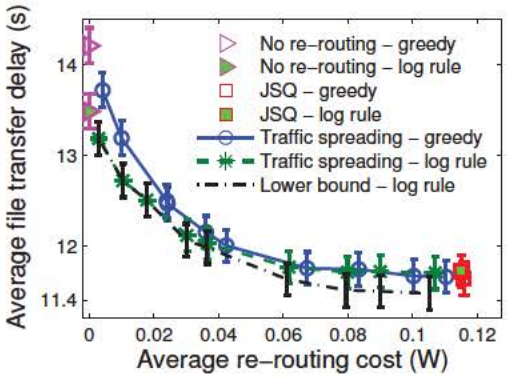
Under this dispatching policy, when a request of user $i$ is generated, user $i$ sends it directly to the BS. Thus there is no additional power expenditure.




(a) Average delay versus weight




(b) Average re-routing cost versus weight




(c) Average delay versus re-routing cost

Figure 5.5: Performance under two-user homogeneous scenarios where $\boldsymbol{\lambda = 0.2}$ arrivals/sec and $\boldsymbol{d = 100}$ m.

#### 5.7.0.2. Join the Shortest Queue (JSQ)

Under this dispatching policy, when there is a new request generated by user $i$, user $i$ sends it to user $j$ that has the least amount of work left. If $i \neq j$, then additional power expenditure occurs.

This work also proposes a lower bound on the average file transfer delay. This bound is obtained from an algorithm consisting of: $i$) using our proposed traffic spreading algorithm at the users; $ii$) modifying the original BS scheduling policy according to Algorithm 5. The motivation behind the lower bound is when the queue of the user with the highest instantaneous data rate is empty, the opportunistic gain will be lost. To recoup the opportunistic gain, the BS can re-route data from other queues to the user with the highest instantaneous data rate. Besides, re-routing a user's own data from other queues saves re-routing cost, while re-routing a user's data to another user requires re-routing cost. Algorithm 5 takes these into account and the re-routing at the BS is forced not to bring obvious additional re-routing cost.

**Algorithm 5** A lower bound of our proposed algorithm

// A modification to the original BS scheduling policy.
// **Definitions:** $\theta_{bs}$: record the re-routed data at the BS.
**Input:** $\boldsymbol{R}(t) = \{R_i^{c_i(t)}, i \in \mathcal{I}, c_i(t) \in \mathcal{S}\}$ : a $1 \times N$ vector of channel-state dependent service rate at time slot $t$
$\boldsymbol{q}(t)$ : a $1 \times N$ vector of queue state
$\boldsymbol{b}(t)$ : a $1 \times N$ vector of the original owner of the head-of-line file of each queue at time slot $t$

$j \leftarrow \arg\max_{i \in \mathcal{I}} R_i^{c_i(t)}$.
**if** $q_j(t)$ is empty **then**
  **if** $\mathcal{K} \equiv \{i : b_i(t) = j, i \in \mathcal{I}\}$ and $\mathcal{K}! = \emptyset$ **then**
    schedule a user $k \in \mathcal{K}$
    $\theta_{bs} = \theta_{bs} -$ served data from user $k$'s queue
    **return**
  **else if** $\mathcal{K} \equiv \{i : b_i(t)! = i, i \in \mathcal{I}\}$ and $\mathcal{K}! = \emptyset$ **then**
    schedule a user $k \in \mathcal{K}$
    **return**
  **else if** $\theta_{bs} < 0$ **then**
    $k \leftarrow \arg\max_{i \in \mathcal{I}} q_i(t)$
    schedule the user $k$
    $\theta_{bs} = \theta_{bs} +$ served data from user $k$'s queue
    **return**
  **end if**
**end if**
**return** execute the original scheduling algorithm

### 5.7.1. The Two-user Scenario

**Homogeneous scenarios:** The simulation results under the greedy and log rule scheduling policies are shown in Figure 5.5 (a)-(c) for a homogeneous scenario where both users are at exactly the same distance from the BS and have identical traffic demands. Figure 5.5 (a) shows that JSQ results in the lowest average delay independent of the scheduling policy, as expected. Under performance-centric case ($\boldsymbol{w} = \mathbf{0}$), traffic spreading can reduce the average delay as much as JSQ does, independent of the scheduling policies. Both JSQ and the traffic spreading have delay improvement up to 18% (greedy) and 14% (log rule) compared to no re-routing. Besides, the gap between our proposed algorithm and the lower bound increases with the decrease of weight. This is because the smaller the weight, the more re-routing at the users will occur. Thus the BS has more opportunities to re-route a user's own data from others' queues to save re-routing cost, and then is able to re-route a user's data to other users. The results in Figure 5.5(b) show that these strategies also correspond to the highest power expenditure. Traffic spreading re-routes as much as JSQ does when $\boldsymbol{w} = \mathbf{0}$. Under energy-sensitive cases ($\boldsymbol{w} > \mathbf{0}$), increasing the weight of re-routing cost results in the energy consumption decreasing rapidly along with increasing average file transfer delay.

Moreover, it can be observed from Figure 5.5 (a) that under large weight ($\boldsymbol{w} \geq \mathbf{200}$), the delay

performance under the log rule scheduler is better than under the greedy scheduler. However, in the performance-centric scenarios, traffic spreading under the greedy scheduler does achieve similar delay performance. In general, the rate of re-routing and the power expenditure is lower under the log rule scheduler. This is because the log rule scheduling policy already tries to balance the queues, while the greedy scheduler does not.

The tradeoff between the average delay and re-routing cost under traffic spreading can be seen from Figure 5.5 (c). It can be seen again that traffic spreading is generally able to achieve the same delay performance at lower power expenditure compared to the greedy scheduler. A very interesting observation indicated by this figure is that most of the delay performance gain can be achieved with a small increase in power expenditure. For example, when $w = \mathbf{0}$, the (maximal) delay performance gain is 18%, and the (maximal) average re-routing cost is $0.12W$. However, when $w = \mathbf{100}$, the delay performance gain is 14% and the re-routing cost is about $0.025W$. This means under the traffic spreading 78% of the maximal performance gain can be achieved with only 20% of the maximal re-routing cost.

The impact of arrival rates on performance is shown in Figure 5.6, where $\boldsymbol{\lambda}$ is scaled, while keeping $d$ and $w$ unchanged. Even at low loads, traffic spreading results in performance gains, with the gains increasing as $\boldsymbol{\lambda}$ increases. For example, when $\boldsymbol{\lambda} = \mathbf{0.175}$ arrivals/sec, the gain is 16%; and it increases to 23% when $\boldsymbol{\lambda} = \mathbf{0.215}$ arrivals/sec. Note that, with the chosen weight, traffic spreading achieves similar delay performance to JSQ at all traffic loads while consuming less than half the extra energy.

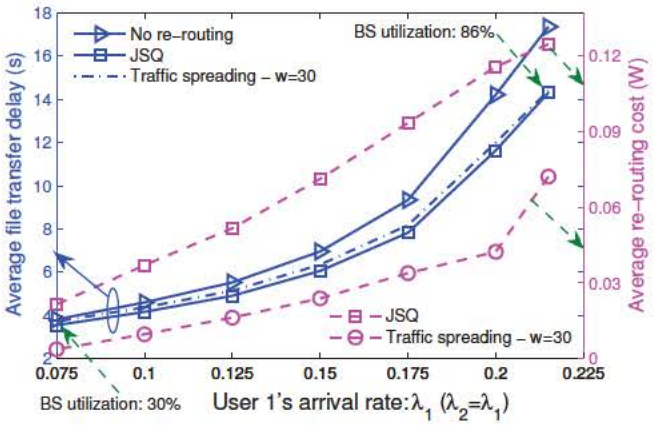



Figure 5.6: Performance vs. arrival rate under two-user homogeneous scenarios where $d = \mathbf{100}$ m and $w = \mathbf{30}$ (confidence intervals are not shown for clarity).

**Heterogeneous scenarios:** Figure 5.7 depicts the delay performance vs. power expenditure tradeoff achieved by traffic spreading in a scenario where one of the users has lower offered traffic as well as a better average channel to the BS. The maximal delay performance gain under traffic spreading is up to 27% (greedy) and 18% (log rule), compared to no re-routing. Similarly to the homogeneous scenario, the average re-routing power expenditure reduces rapidly as the weight is increased from 0 while the average file transfer delay increases much slower. For example,

up to 95% ($w = 10$) of the maximal performance gain can be achieved at only 40% of the maximal re-routing cost. It can be seen again that queue-aware scheduling is indeed beneficial and the combination of the log rule scheduler and traffic spreading is more effective, especially in energy-sensitive scenarios.

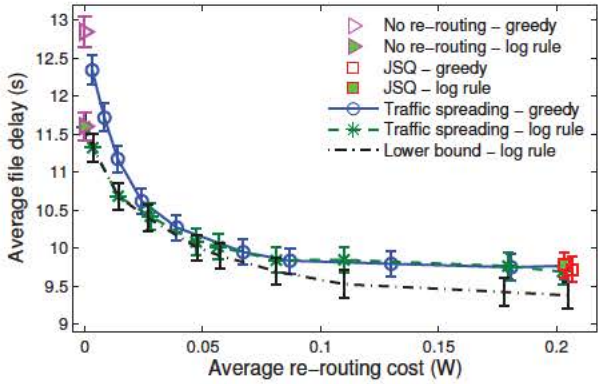



Figure 5.7: Performance vs. re-routing cost under two-user heterogeneous scenarios where $\boldsymbol{\lambda} = \{0.19, 0.21\}$ arrivals/sec and $\boldsymbol{d} = \{92, 100\}$ m.

Figure 5.8 depicts the overall re-routing rate as well as the split between users. Clearly, the performance gain does not originate from simple relaying. In fact, the user with the better average channel and lower traffic ($U_1$) also forwards traffic to the user with the worse channel ($U_2$). $U_1$ does contribute to the bulk of the performance improvement, however it can be observed that $U_2$ forwards a significant amount of traffic for $U_1$ as well. For example, 40% of the total re-routed traffic is from $U_1$ to $U_2$ in the performance-centric scenario with $\boldsymbol{w} = \mathbf{0}$.

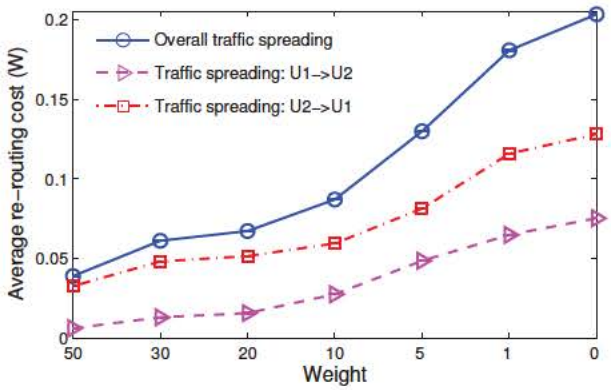
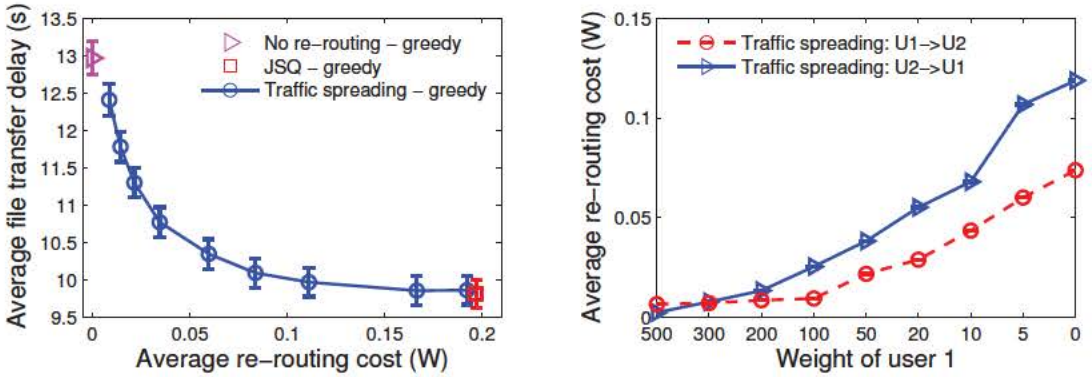



Figure 5.8: Average re-routing rate of each user under greedy scheduler where $\boldsymbol{\lambda} = \{0.19, 0.21\}$ arrivals/sec and $\boldsymbol{d} = \{92, 100\}$m.

The traffic spreading algorithm can also adjust the difference on additional power expenditure among users in heterogeneous scenarios, by assigning different weights to different users. Figure 5.9 depicts the performance versus average power expenditure as well as the average re-routing rates of users in a heterogeneous scenario, where the weight $w_2^1$ changes between 0 and

500 and the weight $w_1^2$ is fixed to 0. It can be seen from Figure 5.9 (b) that as $w_2^1$ increases, the difference of re-routing rates between $U_1 \rightarrow U_2$ and $U_2 \rightarrow U_1$ decreases. The re-routing rate of $U_1 \rightarrow U_2$ also decreases with the increase of $w_2^1$. This is because our traffic spreading aims at balancing users' BS queues. As $w_2^1$ increases, the imbalance of queues is enlarged (the average amount of work in $Q_2$ is larger than that in $Q_1$). Therefore, the chance is reduced that $U_2$ helps $U_1$ re-route its requests to the BS, even $w_1^2 = 0$. The average file transfer delay also increases with the increase of $w_2^1$ as expected, as shown in Figure 5.9 (a).




(a) Average file transfer delay versus re-routing cost

(b) Average re-routing cost of users

Figure 5.9: Fairness under two-user heterogeneous scenarios where $\boldsymbol{\lambda} = \{0.19, 0.21\}$ arrivals/sec and $\boldsymbol{d} = \{92, 100\}$m.

### 5.7.2. Multi-user Scenarios

This subsection presents the performance evaluation results of the proposed heuristic under the greedy scheduler.

**Dynamic programming vs. heuristic performance:** Here first consider a three-user homogeneous scenario. The performance of the proposed heuristic against that of the optimal dispatching policy obtained from solving the dynamic programming formulation are evaluated. It can be observed from Figure 5.10 that the performance of our proposed heuristic is almost as good as that of the optimal dispatching policy, with both able to achieve near identical performance vs. energy consumption tradeoffs. The maximal performance difference between the heuristic and the optimal dispatching policy is less than 2%. The average file transfer delay can be reduced by 18% in this scenario. Similar to the cases considered earlier, it can be seen that up to 60% ($\boldsymbol{w} = \mathbf{30}$) of the maximal performance gain can be achieved at only 20% of the maximal re-routing cost. These results demonstrate that the proposed heuristic is indeed successful in multi-user cases.

**Performance scaling with number of users:** The scaling of the average file transfer delay with the number of users is shown in Figure 5.11 under traffic spreading with different choices of weight, $\boldsymbol{w}$. Here, all users are at the same distance from the BS and offer the same traffic, and the sum arrival rate across all users is fixed to 0.4 arrivals/sec. As would expect, the average delay

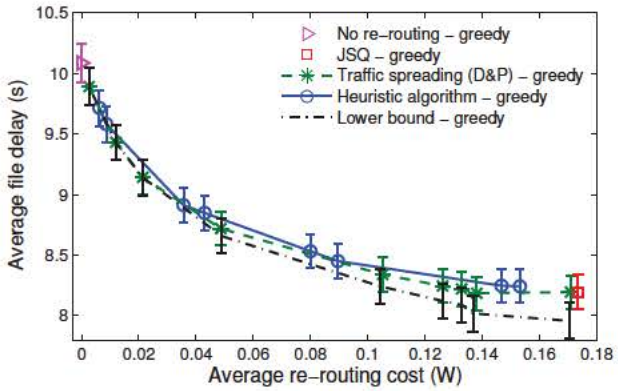



Figure 5.10: Dynamic programming vs. heuristic performance under a three-user scenario where $\sum_{i=1}^{3} \lambda_i = 0.4$ arrivals/sec, $\lambda_i = \lambda_j$ and $d_i = 100$m, $i, j \in \{1, 2, 3\}$.

decreases with the increase number of users due to the increase in overall opportunistic gain. It can also be observed that traffic spreading is able to improve the delay performance by 17% to 19%, compared to no re-routing. As the number of users increases, it can be seen that the gain from traffic spreading first increases slightly and then decreases as the user population increases further. This is due to the fact that opportunistic gain grows slower than linearly with the size of the user population. Note that even in multi-user scenarios, traffic spreading does result in significant performance gains. The power expenditure trends with increasing weight are similar to those of earlier scenarios, with power expenditure reducing steeply with small increase in file transfer delays when the weight, $w$, is increased (energy consumption curves not shown due to space limitations).

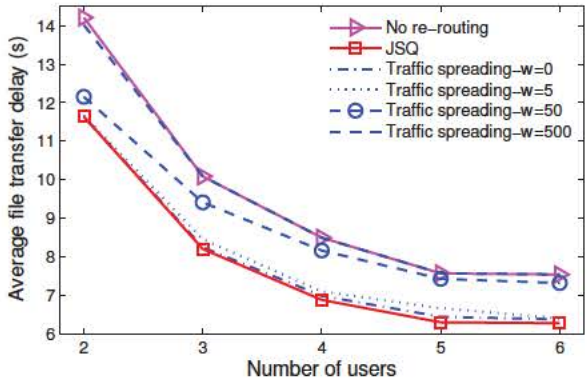



Figure 5.11: Performance versus $N$ users where $\sum_{i=1}^{N} \lambda_i = 0.4$ arrivals/sec, $\lambda_i = \lambda_j$ and $d_i = 100$m, $i, j \in \{1, 2, ..., N\}$.

**Users distributed randomly in a cell of radius 100m:** Simulation results for a scenario where a BS serves users in a service area of radius 100m are presented. Consider instances with four users, each of them distributed uniformly at distances ranging from 10 to 100m (correspond-

ing to average capacity to the BS between 16.0407 to 3.0163Mbps). The rate at which users generate requests is also heterogeneous and chosen uniformly in a range of $0.2 \pm 10\%$ arrivals/sec. For each choice of weight (i.e. performance-energy tradeoff), 50 random instances are evaluated, and the average as well as the $95^{th}$ and $5^{th}$ percentile of the file transfer delay are depicted in Figure 5.12. It can be observed that the average delay performance gain is up to 50%, which is nearly as good as JSQ. This delay performance can be achieved at a re-routing cost that is around half of that under JSQ. Depending on the user requirements, a different tradeoff between power expenditure and delay performance can be chosen. When focus on the $95^{th}$ percentile, the performance gain observed is up to 56%. This demonstrates that traffic spreading is very helpful in improving user performance in instances where overall system performance is poor, which is a very important practical consideration.

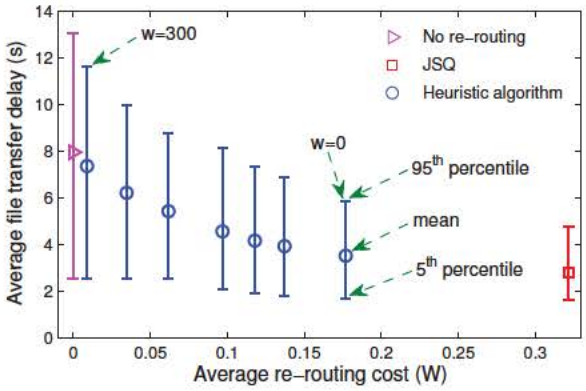


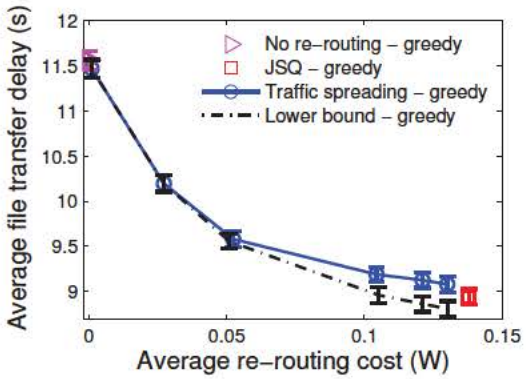

Figure 5.12: Performance under four-user heterogeneous scenarios where $\lambda_i = 0.2 \pm 10\%, d_i \in [10, 100], i \in \{1, 2, 3, 4\}$.

## 5.8. Traffic Spreading in Large Cells

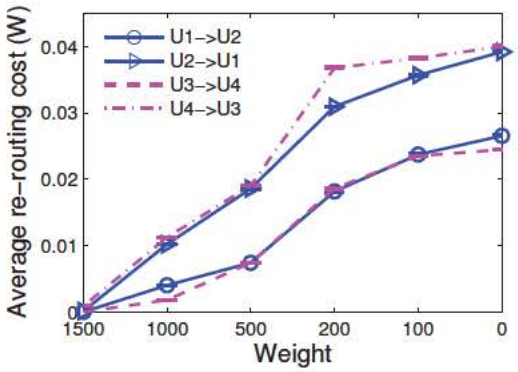
This section extends the proposed traffic spreading algorithm to large cells, where not all users can communicate with each other. It starts by presenting modifications to the system model and formulation in Sec. 5.3 and 5.4, and provide the simulation results.

### 5.8.1. Modification to the system model and formulation

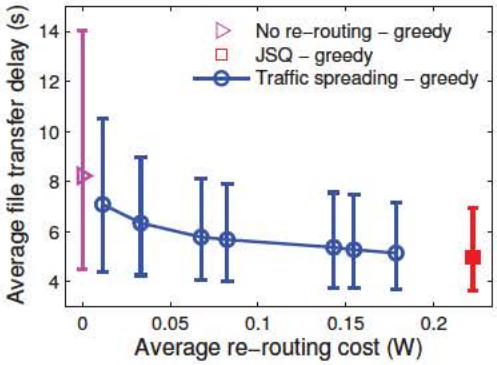
To extend the proposed traffic spreading algorithm to large cells, users are divided into clusters where users in the same cluster can communicate with each other directly. As described in Sec. 5.4, the queue-state-dependent average service rates are indispensable in the dynamic programming formulation. Due to the fact that not all users are within the transmission range of each other in a large cell, these service rates can not be calculated accurately at each user, even through users know the scheduling policy employed by the BS. Therefore, a method is proposed to estimate the average service rates based on users' statistics in each cluster.




(a) Four-user heterogeneous scenario




(b) Average re-routing cost of users




(c) Five users randomly distributed in a cell

Figure 5.13: Performance in a large cell: $(a)$ Performance under a four-user heterogeneous scenario where $\boldsymbol{\lambda} = \{0.13, 0.12, 0.15, 0.14\}$ arrivals/sec and $\boldsymbol{d} = \{100, 92, 86, 78\}$ m; $(b)$ Average re-routing rate of users under the four-user heterogeneous scenario; $(c)$ Five users randomly distributed in a cell where BS transmission power is 0.3 W/MHz, $\lambda_i = 0.13 \pm 10\%$ arrivals/sec, $d_i \in [50, 200]$ m, $i \in \{1, 2, \cdots, 5\}$.

Suppose users in a large cell are divided into $L$ clusters. Let $\mathcal{L}$ denote the set of clusters and $\mathcal{I}_l$ be the set of users in the $l$th cluster. Let $q_{l,i}(t)$ be the queue length of user $i$ and let $s_{l,i}(t)$ be a indicator representing whether user $i$ is served by the BS at slot $t$ or not, where $i \in \mathcal{I}_l, l \in \mathcal{L}$. Moreover, the average time ratio that the BS spends on serving users of cluster $l$ during the time interval $[t_s, t_e]$ is denoted as $\alpha_l$, which is estimated through the following equation

$$\alpha_l = \frac{\sum_{t=t_s}^{t_e} \sum_{i \in \mathcal{I}_l} q_{l,i}(t) \wedge s_{l,i}(t)}{t_e - t_s + 1}. \tag{5.20}$$

The average service rate of user $i$ is given as

$$\mu_i(\boldsymbol{q}) = \alpha_l \cdot \sum_{\boldsymbol{c} \in \boldsymbol{\mathcal{A}}} \left( \prod_{j \in \mathcal{I}} p_j^{c_j} \right) \xi_i(\boldsymbol{q}, \boldsymbol{c}) \cdot R_i^{c_i}, \ \forall i \in \mathcal{I}_l. \tag{5.21}$$

In each cluster, the dispatching policy of users can be calculated by solving an equation similar to (5.15), where the only difference is that the $\mu_i(\boldsymbol{q})$ in (5.15) is now given by (5.21).

### 5.8.2. Performance evaluation

**Four-user heterogeneous scenario:** In this scenario the cell has two clusters ($\mathcal{I}_1$ and $\mathcal{I}_2$), and each cluster has two users ($U_1, U_2 \in \mathcal{I}_1, U_3, U_4 \in \mathcal{I}_2$). Figure 5.13 (a) depicts the tradeoff between performance and additional power expenditure. The maximal delay performance gain under traffic spreading is up to 22%, and up to 80% of the maximal gain is achieved at only 35% of the maximal re-routing cost. The split of average re-routing rates among users is shown in Figure 5.13 (b). It can be observed that users with worse channels ($U_2$ and $U_4$) also re-route files for users with better channels ($U_1$ and $U_3$).

**Users randomly distributed in a cell:** This scenario considers instances with five users, which are distributed uniformly in a cell at distances ranging from 50 to 200m to the BS. In each instance, the two users with the best average channel states act as Cluster Headers (CHs). CHs periodically broadcast their members to neighbouring users. A none-CH user chooses to join a cluster if it can communicate with all the current members of that cluster, augmented with tie-breaking rule. 50 random instances are considered and the average file transfer delay (as well as the $95^{th}$ and $5^{th}$ percentiles) are depicted in Figure 5.13 (c). It is observed that the average delay under no re-routing can be improved by up to $40\%$ through the proposed traffic spreading. Moreover, with traffic spreading, the $95^{th}$ percentile of the delay under no re-routing is improved by up to $50\%$, while up to $73\%$ of the maximal gain can be achieved by at only $18\%$ of the maximal re-routing cost.

## 5.9. Summary

This chapter presented a user-initiated traffic spreading approach, that is transparent to the BS, to improve the downlink delay performance in small cells. The problem of choosing the optimal dispatching policy was formulated as a Markov decision process and studied its properties in a two-user scenario. A heuristic algorithm is also proposed for multi-user scenarios. Our simulation results showed that the proposed approach can improve the delay performance greatly and the bulk of the performance can be achieved with a small increase in power expenditure. Moreover, even in the future when queue-aware scheduling policies are implemented at the BS, our proposed approach can still complement them and improve user performance.

# Chapter 6

# BS-Driven D2D Communication

## 6.1. Introduction

Opportunistic scheduling [22, 23] was proposed for multiuser wireless networks. An opportunistic scheduler exploits the time-varying channels between the base station (BS) and users, to improve the overall system performance. As in the previous chapter, this chapter considers a setting where a BS serves a set of users with stochastic traffic loads, similar to [75, 76, 79, 80, 84, 96]. In such a scenario, the BS at times has no data to transmit to some users and multi-user diversity is reduced. This does not affect performance in large cells (with large user populations) since opportunistic gain scales as a concave function of the number of available users. However, as smaller cells are deployed more densely to increase wireless capacity and meet increasing traffic demands [24, 25], the average number of users in a cell will decrease significantly and a time-varying user population may greatly affect the performance of opportunistic scheduling.

This work proposes a BS-drIven Traffic Spreading (BITS) algorithm that can increase the opportunistic gain in small cells. It considers the downlink that accounts for most of the traffic in a cellular network [77]. The BITS algorithm benefits applications whose performance is sensitive to delay (distribution) of received packets. Such delay-sensitive applications include live streaming, video-conferencing, etc. For instance, in live streaming the video frames which arrive late will be dropped by the video player at the user, thus reducing video quality.

To exploit the user-user communication, BITS leverages the multiple radio interfaces available in most smartphones. BITS uses the cellular link (e.g., 4G) for BS-user communication and a second interface (e.g., WiFi) for user-user communication. The criteria used by BITS are the users' current channel conditions and queue backlogs. Each user measures its perceived channels to other users and shares channel information with the BS. This overhead is manageable as slow fading channels is considered here. BITS takes into account users' backlogs as well as the BS-user and user-user channels, to maximize its scheduling options and hence increase opportunistic gain.

To illustrate the traffic spreading mechanism of BITS, let us consider the example shown in

Figure 6.1, where two users ($U_1$ and $U_2$) are served by a BS. The queues $Q_1$ and $Q_2$ depict the number of packets waiting to be sent to each user. The users perceive similar channel statistics and have similar traffic loads. During each time slot, BITS determines from which queue packets are served and to which user packets are sent (i.e., when and how to spread traffic). In Figure 6.1 (a), the queues are balanced. From Little's law, the average packet delays of $Q_1$ and $Q_2$ are similar (below/above the delay threshold of user's player). Thus traffic spreading is not beneficial and BITS sends packets to the corresponding users directly. In Figure 6.1 (b), there are more packets in $Q_2$ than in $Q_1$ which is nearly empty. This implies the average packet delay of $Q_2$ is higher than that of $Q_1$. Thus in the near future, (more and more) packets in $Q_2$ are likely to become useless when they arrive at the user. BITS reacts to this imbalance in queues and if the channel of $U_1$ is better than $U_2$, BITS sends packets from $Q_2$ to $U_1$, who then forwards them to $U_2$ through the user-user (WiFi) link .

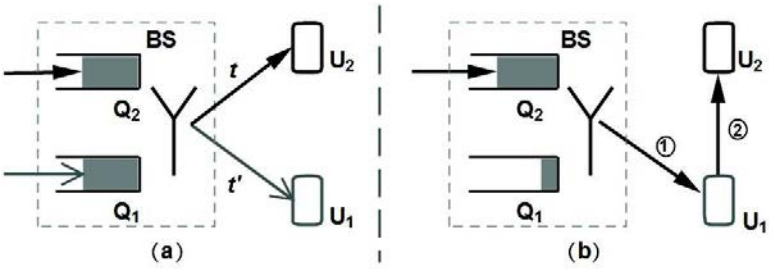



Figure 6.1: An example of BITS: (a) no spreading; (b) with spreading.

The proposed algorithm incurs additional energy consumption due to forwarding traffic among users. As mobile devices have limited energy resources, excessive traffic spreading can result in high penalties in terms of energy consumption. At the same time, balancing energy consumption among the users is an important consideration. This proposed energy-aware BITS algorithm optimizes the degree of spreading for a given energy constraint. The main contributions are summarized as follows:

1. Proposal of an energy-aware scheduling policy (BITS) to increase opportunistic gain by taking into account users' backlogs and the BS-user and user-user channels.

2. The BITS policy is modeled with the objective to maximize delay-sensitive utility under an energy constraint. Using stochastic Lyapunov optimization, an online algorithm is developed and its properties are studied.

3. Evaluation of the BITS using realistic Rayleigh fading channels. Simulation results show that under BITS: *i)* utility is increased greatly and average packet transfer delay is reduced by up to 70% even in homogeneous scenarios; *ii)* in the energy-constrained case (i.e., small energy budget), significant gains (up to 60% of the gain) are typically achieved at only 20% of the energy consumption of performance-centric case (i.e., with a sufficiently large energy budget); *iii)* excellent *fairness* on additional energy consumption among users can

be achieved in heterogeneous scenarios; *iv)* the performance can be improved greatly in scenarios where overall system performance under the proportionally fair scheduling policy used in current 3G and 4G systems is very poor.

4. BITS is evaluated using realistic video traffic traces. Results are provided showing that with BITS, the average Peak Signal-to-Noise Ratio (PSNR) of the received video can be improved by up to 4dB and the frame loss ratio is reduced by up to 90%. Moreover, the quality of the received video varies much more slowly with fluctuations of the wireless channel.

The rest of this paper is organized as follows: the related work is summarized in Section 6.2, followed by the system model and objective optimization in Sections 6.3 and 6.4, respectively. Properties of BITS are discussed in Section 6.5. Performance evaluation in a multi-user system and evaluation using realistic video traces are presented in Section 6.6. Finally, conclusions and future work are provided in Section 6.7.

## 6.2. Related Work

Many scheduling algorithms considering both users' backlogged queues and channel states have been proposed [79, 80, 96]. Among these, [79, 80] propose throughput-optimal MaxWeight and Exponential rules, respectively. Authors in [96] propose the log rule to improve delay performance. All these algorithms react to imbalance in users' queues by sacrificing opportunistic gain in order to balance queues. In contrast, BITS can balance users' backlogs without losing instantaneous gain, by opportunistically exploiting the BS-user and user-user channels.

Another class of scheduling algorithms aims to maximize delay/time-sensitive utility [97, 98], as BITS does. Among these, [98] proposes to maximize delay-sensitive utility to provide delay QoS for each user, while [97] aims to maximize the time average utility to enforce fairness. Compared to our work, the utilities in [97, 98] are defined as concave functions of packet queueing delay, while the utility used in BITS (cf. Section 6.3) is a function of delay-sensitive throughput.

Wang et al. [99] propose a downlink BS-transparent dispatching policy where users spread traffic requests among each other to balance their backlogs. This increases the BS scheduling options and hence improves the performance. Compared to BITS, the dispatching policy is user-initiated and on a per-file basis, while BITS is BS-driven and operates on a per-packet basis. Further, the dynamic programming is used in [99] to determine the optimal dispatching policy and the complexity in large systems is reduced by aggregating users. In BITS, the algorithm is derived from Lyapunov optimization and the complexity is low even in large systems.

Another approach to exploit both the BS-user and user-user channels is opportunistic relaying [82–84, 100]. Among these, [82] proposes the idea of opportunistic relaying and an approach of choosing the best relay that maximizes the minimal quality of BS-relay and relay-user channels. In [83, 84, 100] mobile users themselves, instead of particular relay nodes are used as relays.

The work in [100] considers relaying traffic to areas without cellular coverage and proposes an approach where a user with the best channel to the destination is chosen as the relay. Authors in [83, 84] propose scheduling algorithms to improve the system capacity and fairness. Compared to BITS, [82, 83, 100] assume users have infinitely backlogged queues, which is different from BITS and [84] that consider stochastic traffic loads. Moreover, the delay-sensitive utility as well as delay-energy tradeoff have not been investigated in neither [84] nor the other works discussed above.

## 6.3. System Model

Consider a time-slotted system with $N$ users attached to a BS, where the set of users is denoted as $\mathcal{I} = \{1, 2, ..., N\}$. The BS maintains a separate queue for each user, and denote by $\boldsymbol{Q}(t) \equiv (Q_i(t), i \in \mathcal{I}) \in \mathbb{N}^N$, the number of packets waiting to be sent to each user at slot $t$. Without loss of generality, packet sizes are fixed. The number of packets that can be sent during a slot depends on the modulation scheme. The arrival rates of packets to the BS are modeled through the vector $\boldsymbol{\lambda}(t) = \{\lambda_i(t), i \in \mathcal{I}\}$ where $\lambda_i(t)$ denotes the number of packets that arrive to queue $i$ during slot $t$ and $\lambda_i(t)$ can be arbitrarily bursty. The arrival processes are assumed to be independent across users.

**Channel model:** The channel is time-varying. The channel instances of the BS-user channel at slot $t$ are denoted as $\boldsymbol{c}(t) = \{c_i(t), i \in \mathcal{I}\}$, where $c_i(t)$ is the maximal number of packets that can be sent to user $i$ if the BS chooses to serve user $i$. Assume the BS is aware of users' channel states. The set of all the possible channel instances is $\mathcal{S} = \{c_1^*, c_2^*, \cdots, c_K^*\}$.

**Scheduling policy:** In each slot $t$, the BS scheduler decides from which queue packets are served and to which user the packets are sent. This decision takes into account current queue states and channel states. The scheduling policy is defined through a binary indicator, $\forall i, j \in \mathcal{I}$ and $\forall t$:

$$\sigma_i^j(t) = \begin{cases} 1, & \text{if serving user } j \text{ with packets from queue } i \\ 0, & \text{otherwise} \end{cases}$$

The set of all the possible scheduling policies is defined as

$$\mathcal{C} \equiv \left\{ \boldsymbol{\sigma}(t) : \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{I}} \sigma_i^j(t) = 1,\ \sigma_i^j(t) \in \{0, 1\} \right\}.$$

Assume packets in the same queue are served according to a first-come-first-served discipline and the BS can only serve one user during a slot. The departure rate $\nu_i(t)$ of queue $i$ can be written as

$$\nu_i(t) = \begin{cases} \min[Q_i(t), c_j(t)], & \text{if } \sigma_i^j(t) = 1 \\ 0, & \text{otherwise} \end{cases} \tag{6.1}$$

Finally, the queueing dynamics of the system are

$$Q_i(t+1) = Q_i(t) - \nu_i(t) + \lambda_i(t), \quad \forall i \in \mathcal{I}, \forall t. \tag{6.2}$$

### 6.3.1. Delay-sensitive utility

Let $M_i^t(d_x, d_y)$ denote the number of packets received by user $i$ in slot $t$, with packet delay between $d_x$ and $d_y$. For each user $i$, let $\boldsymbol{d}_i^* = \{d_{i,l}^*, l \in \{1, 2, ..., L\}\}$ be the delay thresholds that determine the utility of a packet. The delay-sensitive utility of user $i$ is defined as

$$\phi(\overline{\mu}_i) = \log(\overline{\mu}_i) = \log \sum_{l=1}^{L-1} w_i^l \lim_{t\to\infty} \frac{\sum_t M_i^t(d_{i,l}^*, d_{i,l+1}^*)}{t} \tag{6.3}$$

where $\overline{\mu}_i$ is the weighted average throughput, $\boldsymbol{w}_i = \{w_i^l, l \in \{1, 2, ...L-1\}\}$ is a weight vector. The non-decreasing and concave logarithmic utility function are used to provide fairness among users. The delay thresholds of different applications may be different. For instance, in live streaming packets that arrive late are dropped at the player and become useless. Under this case, the weight vector can be modelled as $\boldsymbol{w}_i = \{1, 0\}$. In other interactive applications such as gaming, the weights can be modelled by quantizing packet delay and setting the weights in a piecewise constant manner with respect to delay.

**Performance metrics:** The metrics used are the delay-sensitive utility and *re-routing cost*, i.e., the additional energy consumption induced by traffic spreading. Define $\boldsymbol{p}^* = \{p_i^*, i \in \mathcal{I}\}$ as the energy budget per slot (i.e., power) and $\boldsymbol{p}(t) = \{p_i(t), i \in \mathcal{I}\}$ as the re-routing cost in slot $t$, which depends on the scheduling policy $\boldsymbol{\sigma}(t)$.

Our objective is to maximize the sum of users' utilities, subject to users' energy budget:

$$\begin{aligned} \max_{\boldsymbol{\sigma}(t)} \quad & \sum_{i\in\mathcal{I}} \phi(\overline{\mu}_i) \\ \text{s.t.} \quad & \overline{p}_i \le p_i^*, \ \forall i \in \mathcal{I} \end{aligned} \tag{6.4}$$

where $\phi(\overline{\mu}_i)$ is given in (6.3) and $\overline{p}_i$ is the time average of $p_i(t)$ of user $i$.

## 6.4. Stochastic Lyapunov Optimization

Stochastic Lyapunov optimization is used to solve the problem given in (6.4). Since $\phi(\cdot)$ is a concave function, this section first transform the above problem (with functions of time averages) to a problem including only time averages, then uses the drift-plus-penalty framework proposed in [101] to solve it.

### 6.4.1. Problem transformation

The original problem (6.4) is transformed by adding a rectangle constraint and auxiliary variables. Define a rectangle constraint $\mathcal{R} \equiv \{(\mu_1, ..., \mu_N) \in \mathbb{R}^N | 0 \leq \mu_i \leq \gamma_i^{max}, \ \forall i \in \mathcal{I}\}$ where $\gamma_i^{max}$ is a finite constant. Further, denote by $\phi^*$ the maximum utility of problem (6.4), augmented with the constraint $\mathcal{R}$. For each slot $t$, denote by $\boldsymbol{\gamma}(t) = \{\gamma_i(t), i \in \mathcal{I}\}$ a vector of auxiliary variables within the rectangle constraint set $\mathcal{R}$ and assume $\overline{\gamma}_i \leq \overline{\mu}_i, \forall i \in \mathcal{I}$, where $\overline{\gamma}_i$ is the average of $\gamma_i(t)$. According to [101], consider the following transformed problem with only time averages instead of the problem including functions of time averages in (6.4):

$$\max_{\boldsymbol{\sigma}(t)} \quad \sum_{i \in \mathcal{I}} \overline{\phi(\gamma_i)} \tag{6.5}$$

$$\text{s.t.} \quad \overline{p}_i - p_i^* \leq 0, \ \forall i \in \mathcal{I} \tag{6.6}$$

$$\overline{\gamma}_i \leq \overline{\mu}_i, \ \forall i \in \mathcal{I} \tag{6.7}$$

$$\boldsymbol{\gamma}(t) \in \mathcal{R}, \ \forall t \in \{0, 1, 2, ...\} \tag{6.8}$$

where $\overline{\phi(\cdot)}$ is the time average of the utility function $\phi(\cdot)$.

### 6.4.2. Lyapunov optimization

Two virtual queues $\boldsymbol{Z}(t)$ and $\boldsymbol{G}(t)$ as follows are introduced:

$$Z_m(t+1) = \max[Z_m(t) + p_m(t) - p_m^*, 0], \ \forall m \in \mathcal{I} \tag{6.9}$$

$$G_s(t+1) = \max[G_s(t) + \gamma_s(t) - \mu_s(t), 0], \ \forall s \in \mathcal{I} \tag{6.10}$$

Assume $Z_m(0) = 0$, $G_s(0) = 0, \forall m, s \in \mathcal{I}$. The $p_m(t)$ and $\gamma_s(t)$ in (6.9) and (6.10) can be viewed as arrival rates, while $p_m^*$ and $\mu_s(t)$ as departure rates. If the queues $Z_m(t)$ and $G_s(t)$ are stable, i.e., $\lim_{t\to\infty} \mathbb{E}\{Z_m(t)\}/t = 0$ and $\lim_{t\to\infty} \mathbb{E}\{G_s(t)\}/t = 0$, then the constraints in (6.6) and (6.7) can be satisfied [101]. Thus, to ensure that users' additional energy consumption $\overline{p}_s$ is below the energy budget $p_s^*$ and the auxiliary variable $\overline{\gamma}_m$ is within the rectangle constraint set $\mathcal{R}$, the virtual queues $Z_m(t)$ and $G_s(t)$ have to be stable over time. Therefore, define a quadratic Lyapunov function as:

$$L(\boldsymbol{\Theta}(t)) \equiv \frac{1}{2}\left[\sum_{i \in \mathcal{I}} Q_i(t)^2 + \sum_{m \in \mathcal{I}} Z_m(t)^2 + \sum_{s \in \mathcal{I}} G_s(t)^2\right],$$

where $\boldsymbol{\Theta}(t) \equiv [\boldsymbol{Q}(t), \boldsymbol{Z}(t), \boldsymbol{G}(t)]$. For any non-negative constant $V$, define the one-slot Lyapunov drift-plus-penalty as

$$\begin{aligned}\Delta(\boldsymbol{\Theta}(t)) - V\mathbb{E}\{\phi(\boldsymbol{\gamma}(t))\} \equiv &\mathbb{E}\{L(\boldsymbol{\Theta}(t+1)) \\ &- L(\boldsymbol{\Theta}(t))\} - V\mathbb{E}\{\phi(\boldsymbol{\gamma}(t))\}\end{aligned} \tag{6.11}$$

To solve the transformed problem (6.5)-(6.8), then need to minimize (6.11) in each slot [101]. The intuitions behind this are $i$) by minimizing the drift $\Delta(\mathbf{\Theta}(t))$, the virtual queues $Z_m(t)$ and $G_s(t)$ will be stable and thus the constraints in (6.6) and (6.7) are satisfied; $ii$) similarly, (6.5) is solved by minimizing the penalty $-V\mathbb{E}\{\phi(\boldsymbol{\gamma}(t))\}$. This relationship is given by Theorem 3 presented in Sec. 6.4.4. For the property of the drift-plus-penalty defined in (6.11), it has the following property:

**Lemma 2.** *For all possible values of $\mathbf{\Theta}(t)$ and under any scheduling policy $\boldsymbol{\sigma}(t) \in \mathcal{C}$, the drift-plus-penalty has the following upper bound for all slots $t$:*

$$\begin{aligned}\Delta(\mathbf{\Theta}(t)) - V\mathbb{E}\{\phi(\boldsymbol{\gamma}(t))|\mathbf{\Theta}(t)\} \leq D - V\mathbb{E}\{\phi(\boldsymbol{\gamma}(t))|\mathbf{\Theta}(t)\} \\ + \sum_{m\in\mathcal{I}} Z_m(t)\mathbb{E}\{(p_m(t) - p_m^*)|\mathbf{\Theta}(t)\} \\ + \sum_{s\in\mathcal{I}} G_s(t)\mathbb{E}\{(\gamma_s(t) - \mu_s(t))|\mathbf{\Theta}(t)\} \\ + \sum_{i\in\mathcal{I}} Q_i(t)\mathbb{E}\{(\lambda_i(t) - \nu_i(t))|\mathbf{\Theta}(t)\} \end{aligned} \tag{6.12}$$

*where* $D \equiv \frac{1}{2}\sum_{i\in\mathcal{I}}[(\overline{\lambda}_i)^2 + (\overline{\nu}_i)^2] + \frac{1}{2}\sum_{s\in\mathcal{I}}[(\overline{\gamma}_s)^2 + (\overline{\mu}_s)^2] + \frac{1}{2}\sum_{m\in\mathcal{I}}[(\overline{p}_m)^2 + (p_m^*)^2]$.

*Proof*: From the queueing dynamics (6.2), then

$$\begin{aligned}Q_i(t+1)^2 - Q_i(t)^2 &= (Q_i(t) - \nu_i(t) + \lambda_i(t))^2 - Q_i(t)^2 \\ &\leq \nu_i(t)^2 + \lambda_i(t)^2 + 2Q_i(t)(\lambda_i(t) - \nu_i(t)),\ \forall i \in \mathcal{I}\end{aligned}$$

Similarly,

$$\begin{aligned}Z_m(t+1)^2 - Z_m(t)^2 \leq p_m(t)^2 + (p_m^*)^2 \\ + 2Z_m(t)(p_m(t) - p_m^*),\ \forall m \in \mathcal{I} \\ G_s(t+1)^2 - G_s(t)^2 \leq \gamma_s(t)^2 + \mu_s(t)^2 \\ + 2G_s(t)(\gamma_s(t) - \mu_s(t)),\ \forall s \in \mathcal{I}\end{aligned}$$

Taking conditional expectations of the above three equations and summing over $i, m, s \in \mathcal{I}$, a bound on $\Delta(\mathbf{\Theta}(t))$ is obtained. The lemma is then proved by subtracting $V\mathbb{E}\{\phi(\boldsymbol{\gamma}(t))|\mathbf{\Theta}(t)\}$ from both sides. ■

### 6.4.3. Proposed BITS Algorithm

The BITS algorithm seeks to minimize the upper bound of (6.12) instead of directly minimizing the drift-plus-penalty itself. As shown in Sec. 6.4.4, this does not affect the optimality of the solution. The algorithm works as follows:

*1) Auxiliary variables:* Based on $\boldsymbol{G}(t)$, choose $\boldsymbol{\gamma}(t)$ in each slot $t$ such that the following function is maximized

$$\max_{\boldsymbol{\gamma}(t)} \quad V\sum_{s\in\mathcal{I}}\phi(\gamma_s(t)) - \sum_{s\in\mathcal{I}} G_s(t)\gamma_s(t) \tag{6.13}$$

$$\text{s.t.} \quad 0 \le \gamma_s(t) \le \gamma_s^{max},\ \forall s \in \mathcal{I} \tag{6.14}$$

Since the auxiliary variables $\boldsymbol{\gamma}(t)$ are independent, the above maximization can be decoupled as maximizing $V\phi(\gamma_s(t)) - G_s(t)\gamma_s(t), \forall s \in \mathcal{I}$, subject to $0 \le \gamma_s(t) \le \gamma_s^{max}$. The peak value of this objective function is obtained when $\gamma_s(t) = V/G_s(t)$ for $G_s(t) > 0$. Therefore, by taking into account the constraint of $\gamma_s(t)$ in (6.14), the following optimal solution to the above problem is obtained:

$$\gamma_s(t) = \begin{cases} \frac{V}{G_s(t)}, & G_s(t) \ge \frac{V}{\gamma_s^{max}} \\ \gamma_s^{max}, & G_s(t) < \frac{V}{\gamma_s^{max}} \end{cases}$$

The value of $G_s(t)$ directly affects the value of $\gamma_s(t)$. If the value of $G_s(t)$ is small, this implies that the time average of $\gamma_s(t)$ is very close to that of $\mu_s(t)$, which enforces the stability of virtual queue $G_s(t)$. If the value of $G_s(t)$ is large, then a small $\gamma_s(t)$ should be chosen to enforce queue stability. The complexity of (6.13) is $O(N)$.

*2) Scheduling policy:* Based on $\boldsymbol{Q}(t)$, $\boldsymbol{d}(t)$, $\boldsymbol{G}(t)$, $\boldsymbol{Z}(t)$ and $\boldsymbol{c}(t)$, choose $\boldsymbol{\sigma}(t) \in \mathcal{C}$ in each slot $t$ to minimize

$$\begin{aligned}\sum_{m\in\mathcal{I}} Z_m(t)p_m(\boldsymbol{\sigma}(t)) - \sum_{i\in\mathcal{I}} Q_i(t)\nu_i(\boldsymbol{\sigma}(t)) \\ - \sum_{s\in\mathcal{I}} G_s(t)\mu_s(\boldsymbol{\sigma}(t))\end{aligned} \tag{6.15}$$

The $Z_m(t), Q_i(t)$ and $G_s(t)$ in (6.15) can be interpreted as the weights of re-routing cost, service rate and delay-sensitive throughput, respectively. Re-routing is only beneficial if the cost of re-routing is low. For this reason, BITS only allows re-routing if the instantaneous channel quality between the two users is good and the forwarding incurs no additional packet queuing delay at the users. Under a large energy budget $p_i^*$, the value of $Z_m(t)$ is always equal to zero according to (6.9). Thus, in each slot the scheduling policy is to balance the queues $\boldsymbol{Q}(t)$ and $\boldsymbol{G}(t)$ among users. Under a small energy budget, the value of $Z_m(t)$ is no longer always equal to zero. The scheduling policy starts to trade off between the opportunistic gain and the re-routing energy cost. The complexity of this policy is $O(N^2)$ because it chooses $\boldsymbol{\sigma}(t) \in \mathcal{C}$ to minimize (6.15) in each slot and there are $N^2$ different $\boldsymbol{\sigma}(t)$.

*3) Queue updates:* Update the virtual queues $\boldsymbol{Z}(t)$ and $\boldsymbol{G}(t)$ according to (6.9) and (6.10) and the $\boldsymbol{\gamma}(t), \boldsymbol{p}(t)$ determined from previous two steps. The queue $\boldsymbol{Q}(t)$ is updated according to (6.2) and the scheduling decision $\boldsymbol{\sigma}(t)$.

### 6.4.4. Optimality analysis

To prove the optimality of BITS, thus subsection show that the difference between the utility under BITS and the optimal utility can be made arbitrarily small:

**Theorem 3.** *Assume initially all the queues $\boldsymbol{Q}(t)$, $\boldsymbol{Z}(t)$ and $\boldsymbol{G}(t)$ are empty. For a particular constant $V > 0$, the achieved time-average utility under the BITS algorithm satisfies:*

$$\lim_{t\to\infty} \inf \sum_{i\in\mathcal{I}} \phi(\overline{\mu}_i) \geq \phi^* - \frac{D}{V} \tag{6.16}$$

*where $\phi^*$ is the maximal achievable utility under all possible scheduling policies $\boldsymbol{\sigma}(t) \in \mathcal{C}$.*

Theorem 3 indicates that by increasing $V$, the utility under BITS can be made arbitrary close to the optimal utility. Note that a large $V$ also results in large average backlogs of the virtual queues $\boldsymbol{Q}(t)$ and $\boldsymbol{Z}(t)$. The theorem can be proved by first proving the existence of a stationary queue-state-unaware scheduling decision, followed by inserting the decision into (6.12) to remove the dependence on $\boldsymbol{\Theta}(t)$ in the expectations. After that, the theorem can be proved by applying iteration over slots, using the property of Jensen's inequality and then by taking the limit (similar to the proof Theorem 5.1 in [101]).

## 6.5. Property of the BITS Scheduling Policy

This section studies the properties of BITS in a two-user two-channel-state system. The channel is a Markovian channel with *on* and *off* states. Consider a homogeneous scenario where the transition probabilities of the BS-user link are $p^i_{on2on} = p^i_{off2off} = 0.8$; $p^i_{on2off} = p^i_{off2on} = 0.2, \forall i \in \mathcal{I}$. The packet size is fixed to 1. Ausume one packet can be sent in a slot (1ms) if the channel is *on* and zero packets is sent if the channel is *off*. The arrivals are according to a Poisson process with average arrival rates $\boldsymbol{\lambda} = \{0.3, 0.3\}$ packets/ms. For each user $i$, the delay threshold vector $\boldsymbol{d}^*_i = \{0, 20, +\infty\}$ms and the weight $\boldsymbol{w}_i = \{1, 0\}$. The additional energy cost $p_i(t)$ is 1 mJ for re-routing one packet, $\forall i \in \mathcal{I}$. For simplicity, here assume that the user-user link has no forwarding delay. (*In Section 6.6, however, forwarding delay is considered.*)

Let $h_i$ denote the queueing delay of head-of-line packet of $Q_i, \forall i \in \mathcal{I}$. Further, define the *combined queue* as follows:

$$U_i\text{'s combined queue} = \begin{cases} Q_i + G_i, & h_i < d^*_i \\ Q_i, & \text{otherwise} \end{cases} \tag{6.17}$$

The scheduling decisions of BITS is shown in Figure 6.2 and 6.3. The axes represent the lengths of combined queues and the figures depict scheduling decisions at each queue state (except for Figure 6.3(d)). To help distinguish between different scheduling decisions, the diagonal where the lengths of combined queues are equal is drawn. The following properties can be observed:

**Instantaneous gain vs. queue balancing:** BITS can exploit instantaneous gain and/or balance queues based on current channel states, packet delays, actual and virtual queue states. Exploiting instantaneous gain is achieved by sending packets that contribute the most to the utility, while balancing queues aims to reduce the future number of packets that have no contribution to the utility. This property is presented in detail by considering different combinations of channels states.

*1) Both channels are on:* There is no re-routing under this case, i.e., BITS always serves users with their own packets. Thus there are two scheduling decisions, i.e., to serve $Q_1$ or $Q_2$, as shown in Figure 6.2. It can be observed that the scheduling policies are separated by the diagonal, which implies that BITS always balances the combined queues. This can be derived from (6.15) where $p_l(\boldsymbol{\sigma}(t))$ is always equal to zero, $\forall l \in \mathcal{I}$. It can be further observed for different values of $h_1$ and $h_2$:

- $h_1, h_2 < 20$ms: as shown in Figure 6.2 (a), BITS balances the combined queues, aiming to keep in the future as many packets that contribute to the utility as possible.

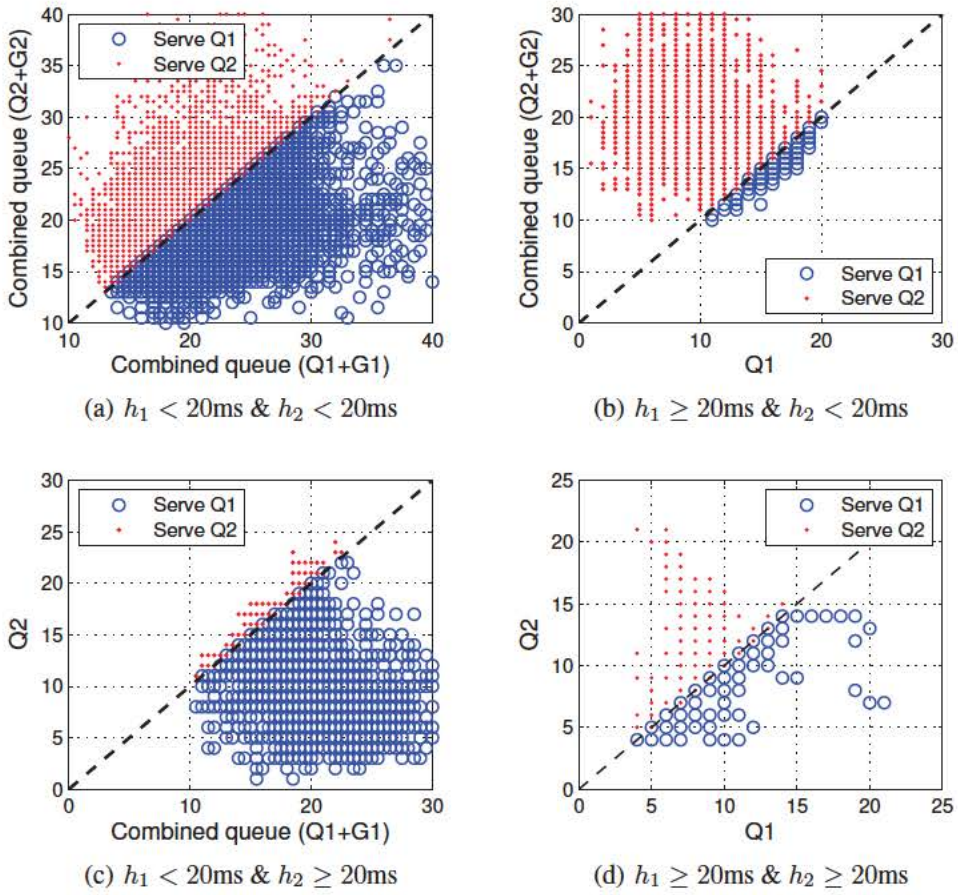
- $h_1 \geq 20$ms$, h_2 < 20$ms: The head-of-line packet of $Q_1$ under this case does not contribute to the utility, thus from (6.17) it can be known that $U_1$'s combined queue is $Q_1$. Again, BITS algorithm balances the combined queues, i.e., $Q_1$ and $Q_2 + G_2$, as shown in Figure 6.2 (b). This implies even $Q_2$ is smaller than $Q_1$, the policy may still serve $U_2$ to exploit instantaneous gain (note that $G_2 \geq 0$).

- $h_1 < 20$ms & $h_2 \geq 20$ms: similar to the previous case.

- $h_1, h_2 \geq 20$ms: The head-of-line packets of both users do not contribute to the utility, thus BITS only balances the actual queues $Q_1$ and $Q_2$, as shown in Figure 6.2 (d).

*2) One user's channel is on, the other's is off:* Without loss of generality, let us focus on the case where $U_1$'s channel is *on* and $U_2$'s is *off*. Note that there is re-routing from $Q_2$ to $U_1$ if the energy budget permits. The scheduling decisions is depicted in Figure 6.3 where the decisions are shown when both $h_1$ and $h_2$ are smaller than 20ms. Scheduling decisions under other values are similar. It can be observed that under large energy budget (e.g., $p_i^* = 1$W·h (over a second),$\forall i \in \mathcal{I}$), BITS still balances the combined queues. However, when energy budget decreases, BITS reacts less to the imbalance in combined queues. This is explained in depth through another property of BITS.

**Performance vs. energy consumption:** The tradeoff between performance and energy consumption can be seen clearly from Figure 6.3. The scheduling decisions are depicted when both $h_1$ and $h_2$ are smaller than 20ms, $U_1$'s channel is *on* and $U_2$'s channel is *off*. If BITS chooses to serve $Q_2$, there will be re-routing from $Q_2$ to $U_1$. Given a large energy budget, e.g., $p_i^* = 1$W·h, BITS behaves exactly the same as in Figure 6.2 (a), i.e., balancing combined queues. The re-routing area ($Q_2 \rightarrow U_1$) diminishes progressively with the decrease of energy budget, as shown in Figure 6.3 (a)-(c). This implies that performance is sacrificed in order to reduce the energy




(a) $h_1 < 20$ms & $h_2 < 20$ms

(b) $h_1 \geq 20$ms & $h_2 < 20$ms

(c) $h_1 < 20$ms & $h_2 \geq 20$ms

(d) $h_1 \geq 20$ms & $h_2 \geq 20$ms

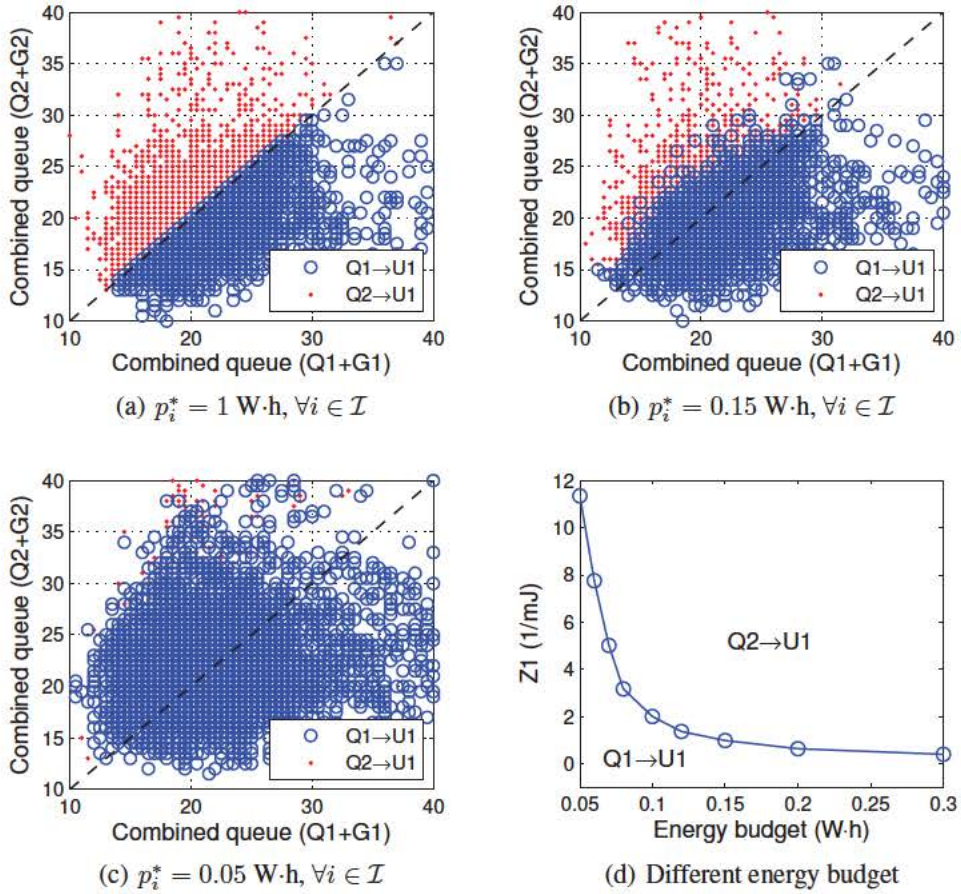
Figure 6.2: Properties of BITS: both users' channels are *on*.

cost. When the energy budget is very small, re-routing only occurs when the imbalance between users' combined queues is very large. Furthermore, from (6.15) it can be derived that if the difference between the combined queues of $U_2$ and $U_1$ at time slot $t$ is larger than the length of virtual queue $Z_1(t)$, BITS will choose to send packets from $Q_2$. Based on this derivation, the scheduling decisions is depicted in Figure 6.3 (d) where the axes are energy budget and length of $Z_1$, respectively. In this figure, the scheduling decision above the curve is re-routing from $Q_2$ to $U_1$, and below the curve is no re-routing. It can be observed clearly that the re-routing degree increases with the increase of energy budget, which shows the tradeoff between performance and energy consumption. Another very interesting observation is that the chances of re-routing from $Q_2$ to $U_1$ can be increased greatly with even a small increase in energy budget.

## 6.6. Performance Evaluation

To evaluate BITS, it is compared with following policies:

1. *Queue-unaware, proportionally fair scheduling (PF)*: At any slot $t$, a PF scheduler chooses




(a) $p_i^* = 1$ W·h, $\forall i \in \mathcal{I}$ (b) $p_i^* = 0.15$ W·h, $\forall i \in \mathcal{I}$

(c) $p_i^* = 0.05$ W·h, $\forall i \in \mathcal{I}$ (d) Different energy budget

Figure 6.3: Properties of BITS: $U_1$'s channel is *on*, $U_2$'s channel is *off*, $h_1 < 20$ms and $h_2 < 20$ms

to serve the user $i$ with the maximum $R_i(t)/R_i'(t)$, where $R_i(t)$ is $U_i$'s instantaneous data rate and $R_i'(t)$ is the exponentially smoothed average service rate of $U_i$ [76].

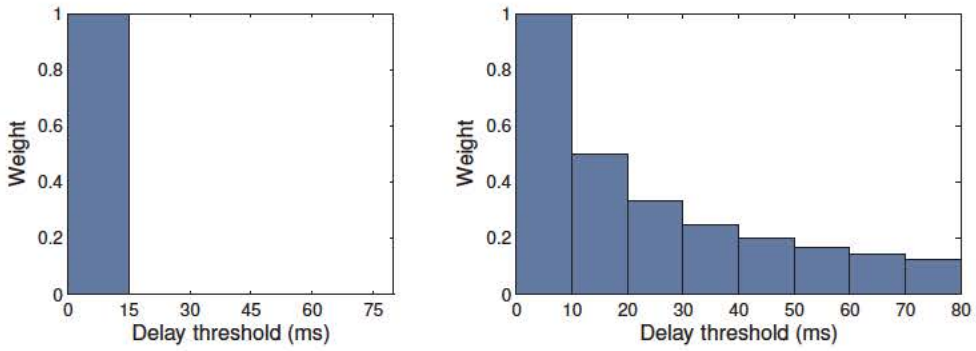
2. *Queue-aware, log rule scheduling*: Queue-aware means the scheduler is aware of the queue length. At time $t$, a log rule scheduler makes decisions based on current channel state and the logarithm of queue length [96].

3. *Queue-aware, maximal re-routing (MaxRR)*: At time $t$, a MaxRR scheduler chooses to serve user $i$ that has the largest instantaneous data rate with packets from the longest queue. If the packets do not belong to user $i$, they will be forwarded by user $i$ to the corresponding user through user-user link.

Note that among the above scheduling policies, re-routing only occurs under MaxRR.

### 6.6.1. Simulation setup

**BS-user link:** Time is slotted and each slot lasts for 1ms. The BS-user channel is a Rayleigh fading channel where the Signal-to-Noise-Ratio (SNR) is assumed to be constant during each slot.




(a) Type I (b) Type II

Figure 6.4: Delay thresholds and the corresponding weights.

Other channel settings are listed in Table 6.1 (note that the specific choice of parameters has no significant impact on the fundamental tradeoff between performance improvement and re-routing energy consumption). Modulation scheme to SNR according to the mechanism specified in [102] are adopted. The setting of the parameters in the PF and log rule scheduling policies are taken from [76] and [96], respectively.

**User-user link:** The user-user channels are also Rayleigh fading channels where the path loss exponent and Doppler shift are set according to Table 6.1. The bandwidth is 20MHz. The durations of aSlotTime, SIFS and DIFS are $9\mu s$, $10\mu s$ and $28\mu s$, respectively. The minimal and maximal contention window are set to 15 and 1023, respectively. The RTS/CTS mechanism is disabled. The lengths of PHY header, MAC header and ACK are 192bits, 256bits and 304bits. The setting of data rate with respect to SNR is according to [103].

Two different types of delay thresholds (Type I and Type II) are evaluated. In Type I, the thresholds and corresponding weights are shown in Figure 6.4 (a). Type I can represent a type of applications where packets are dropped if their delays exceed certain value, e.g., live video streaming. The relationship between thresholds and the corresponding weights of Type II is shown in Figure 6.4 (b). Type II can represent a type of interactive applications, e.g., gaming.

Without loss of generality, the packet size is fixed to 175 bytes unless otherwise specified.

Table 6.1: CHANNEL PARAMETERS OF BS-USER LINK

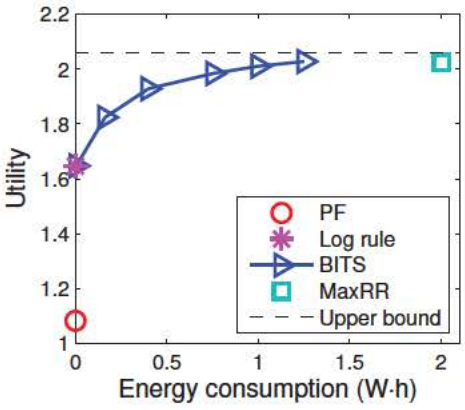
| Parameters | Value |
|---|---|
| Bandwidth | 5 MHz |
| BS Tx power | 0.1/5 W/MHz |
| Noise spectral density | $10^{-8}$/5 W/MHz |
| Path loss exponent (Urban Area) | 3 |
| Doppler shift (ITU Pedestrian A) | 5 Hz |

Assume the additional energy cost $p_i(t)$ for re-routing a packet is 1 mJ, $\forall i \in \mathcal{I}$. The constant $V$ in (6.11) is set to 100. Assume during each slot, $M(t) \in \mathbb{Z}_+$ packets are transmitted, depending on the value of instantaneous SNR. In the simulation results, the values of energy budget and energy consumption of users are given within a second (i.e., power).

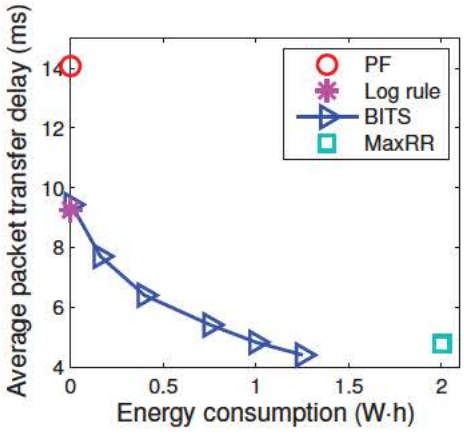
### 6.6.2. The two-user scenarios

**Homogeneous scenarios:** The simulation results are shown in Figure 6.5 (due to space limitation, the results under the Type II threshold are not shown). The x-axes are energy consumption while the y-axes are utility and average packet transfer delay, respectively. The tradeoff between performance and energy budget under BITS can be clearly seen in both Figure 6.5 (a) and (b). Note that the upper bound in Figure 6.5 (a) is obtained by assuming all the packets are transmitted without delay. Under large energy budgets, the performance of BITS can be as good as that under maximal re-routing. Compared to PF, the utility can be increased by 1 under both BITS and MaxRR. The average packet transfer delay with BITS can be reduced by up to 72%. To provide further insights into these results, in Figure 6.5 (c) the Cumulative Distribution Function (CDF) of packet delay is plotted. It can be observed that under BITS (with a large energy budget) and MaxRR, almost all packets are served with a delay below 15ms, while under the log rule and PF, around 20% and 40% of the packets have delays exceeding 15ms. Moreover, the worst-case packet delays under BITS and MaxRR are less than 45ms, while these delays under log rule and PF are around 60ms and 75ms, respectively. Under small energy budget, the re-routing energy consumption decreases rapidly along with the decrease of utility and the increase of average packet transfer delay, as shown in Figure 6.5 (a) and (b). Further, it can be observed that BITS is able to achieve the same performance as MaxRR while consuming less energy (65%). This is because BITS can optimize the degree of re-routing under a specified energy budget. An another interesting observation is that most of the performance gain under BITS can be achieved at small increase in energy consumption, e.g., 70% of the utility gain can be achieved at only 30% of the maximal energy consumption under BITS.

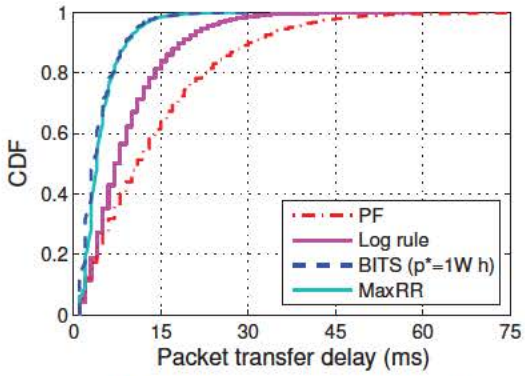
The impact of arrival rates on the utility and delay performance under Type II threshold is shown in Figure 6.6, where $\boldsymbol{\lambda}$ is scaled while keeping other parameters unchanged. As expected, the average transfer delay increases with the increase of arrival rates, as indicated in Figure 6.6 (b). It can also be observede that when the arrival rates are low (e.g., 1.4 packets/ms), the utilities of all the policies are almost the same. This is because almost all the packets can be served with low queueing delay at the BS. As the packet arrival rates increase, packet queueing delays increase and more and more packets are going to exceed the delay threshold. This is why the utilities of all the scheduling policies first increase and then decrease with the increase of packet arrival rate. However, since BITS and MaxRR exploit the local user-user communication to spread traffic, the utilities under these policies decrease much slower than those under the log rule and PF. Note that for a large energy budget, the performance of BITS is even better than MaxRR while consuming less energy for re-routing. This is because BITS is aware of the packet delays, and thus can make




(a) Average utility (Type I)




(b) Average packet delay (Type I)



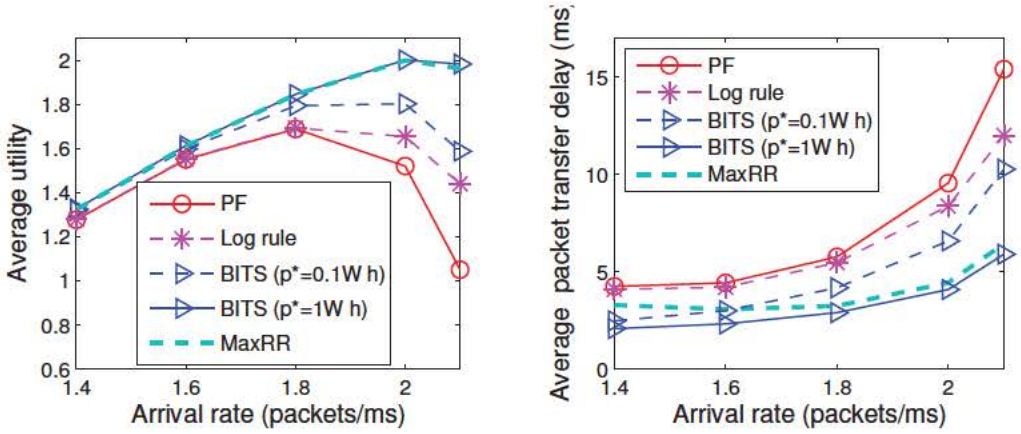

(c) CDF of the packet delay (Type I)

Figure 6.5: Two-user homogeneous scenario: the average $\mathbf{SNR}^{\text{bs-user}} = \{9, 9\}$dB, $\text{SNR}_{ij}^{\text{user-user}} = 9$dB, $\boldsymbol{\lambda} = \{2, 2\}$packets/ms, $i \neq j$.

better scheduling decisions than MaxRR.

**Heterogeneous scenarios:** Figure 6.7 depicts the tradeoff between performance and energy consumption achieved by BITS under a scenario where one user has a lower traffic load as well as a worse average BS-user channel quality. The upper bound also comes from the assumption as in the homogeneous scenarios that all the packets are transmitted without delay. The maximal utility and delay improvements under BITS are 0.7 and 75%, respectively, compared to PF. When the energy budget is large, the delay performance of BITS is even better than MaxRR, while requiring only 65% of the energy consumption. This is because BITS introduces less additional delay than MaxRR, where the delay comes from packet forwarding among users. Thus, in Figure 6.7 (b) it can be observed that the average packet transfer delay under BITS (with a large energy budget) is lower than that under MaxRR. Similar to the homogeneous scenario, the re-routing energy consumption reduces rapidly as the energy budget decreases, while the performance decreases slowly.

The overall re-routing energy consumption as well as the split between users under BITS is



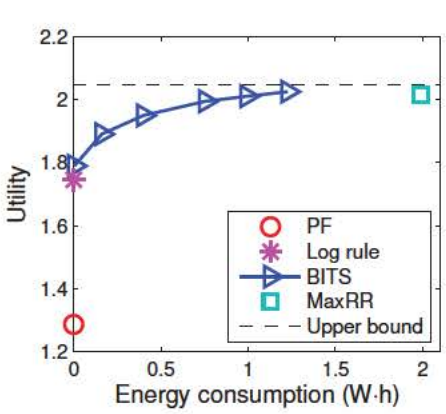

(a) Average utility (Type II) (b) Average delay (Type II)

Figure 6.6: Two-user homogeneous scenarios with different arrival rates: the average $\mathbf{SNR}^{\text{bs-user}} = \{9, 9\}$dB, $\text{SNR}_{ij}^{\text{uses-user}} = 9$dB, $i \neq j$.

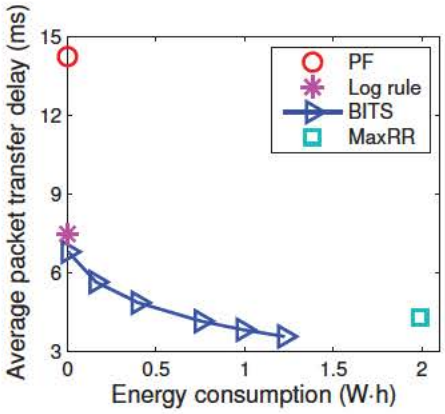
shown in Figure 6.7 (c). It can be observed that both of the two users contribute to the performance improvement, i.e., even the user with a worse channel ($U_2$) forwards packets (thus consuming energy) to the user that has a good channel ($U_1$). Another observation is that the re-routing energy consumption is always constrained by the energy budget, as expected from the theoretical analysis. Further, when energy budget is below 0.4W·h, the energy consumptions of the two users are the same, both are equal to the budget. This implies that BITS can balance the energy consumption among users when given the same energy budget for the users. It can also be observed that when the energy budget is equal to or above 0.4W·h, the re-routing energy consumption of $U_1$ is slightly higher than that of $U_2$, while both of them are below the energy budget. This is because the average channel quality of $U_1$ is better than that of $U_2$, so $U_1$ has more opportunities to forward packets for $U_2$. Since the energy budget is large, BITS will exploit as much as possible these opportunities to maximize the performance improvement.

### 6.6.3. Multi-user scenarios

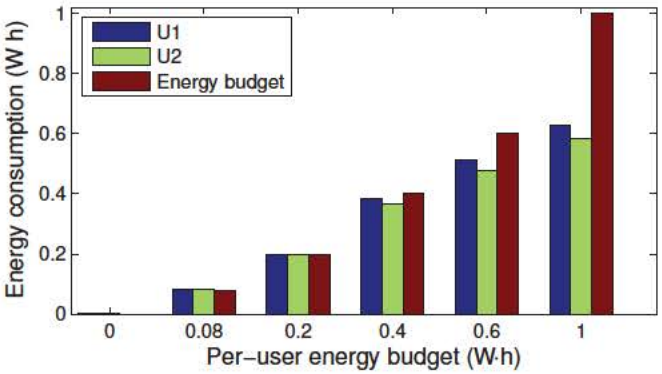
**Homogeneous scenarios (performance scaling with number of users):** The scaling of the utility with the number of users is shown in Figure 6.8. Here, all users have the same traffic load, and the sum of arrival rates across all users is fixed to 4 packets/ms. The average SNRs of the BS-user and user-user channels are the same. All users can communicate with each other and each user has a sufficiently large energy budget. An interesting observation from Figure 6.8 (a) is that the utility decreases when the number of users increases. Another observation in Figure 6.8 (a) is that the difference between utilities under BITS and PF increases with an increasing number of users. The reason behind this is as the user population increases, the BS have more and more choices to spread traffic among users. This can be seen more clearly from Figure 6.8 (b) where the upper bound of the utility is fixed to 1 for all numbers of users and depict the differences




(a) Average utility (Type II)




(b) Average delay (Type II)



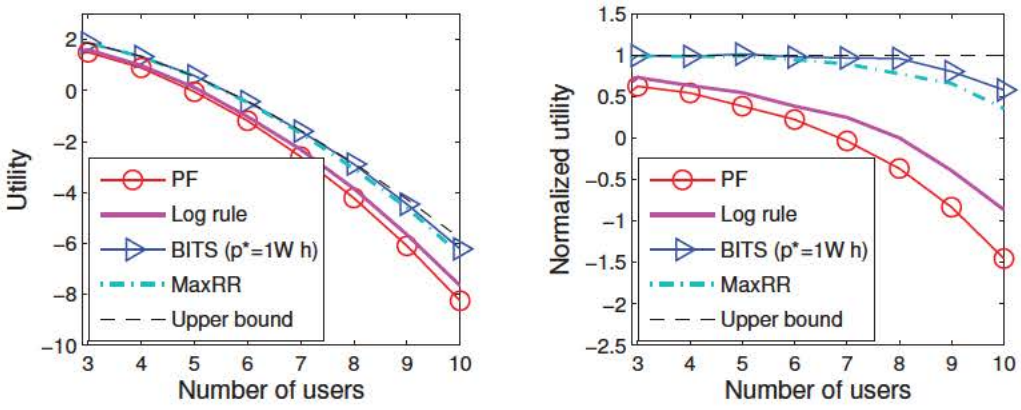

(c) Per-user energy consumption under BITS (Type II)

Figure 6.7: Two-user heterogeneous scenario: the average $\mathbf{SNR}^{\text{bs-user}} = \{10, 9\}$dB, $\text{SNR}_{ij}^{\text{uses-user}} = 9$dB, $\boldsymbol{\lambda} = \{2.2, 1.8\}$packets/ms.

between the upper bound and utilities under all the scheduling algorithms.

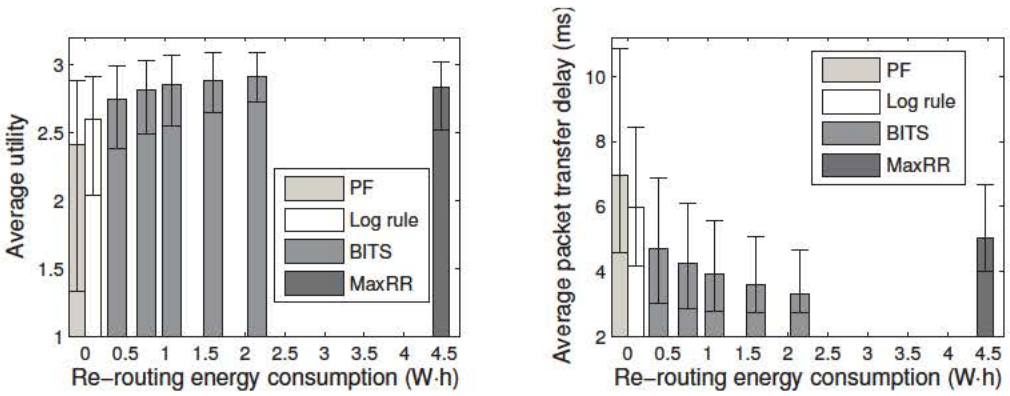
**Heterogeneous scenarios (users randomly distributed in a cell):** Results where users are served by a BS with a service range of 100m are presented. Instances with four users are considered, all of them randomly distributed at distances ranging from 50m to 100m to the BS (corresponding to average SNRs between 19dB and 10dB). Note that if the channel between two users is very poor, the scheduler will not spread traffic among them. Users' arrival rates are also heterogeneous and the arrival rate for each user is chosen randomly in a range of $1.25 \pm 10\%$ packets/ms. 100 instance are evaluated for each chosen re-routing energy budget. The average results as well as the $95^{th}$ and $5^{th}$ percentile of the utility and average packet transfer delay are shown in Figure 6.9.

It can be observed that under BITS, average utility can be improved by 0.4 and the average packet transfer delay can be reduced by up to 50%, compared to PF. Even compared to MaxRR, the delay under BITS can be improved by up to 35% while having only half of the energy consumption. This is due to the smart control of traffic spreading of BITS as well as the large differ-




(a) Utility (Type II) (b) Normalized utility (Type II)

Figure 6.8: Homogeneous scenarios with different number of users: the average $\mathbf{SNR}^{\text{bs-user}} = \{9, 9\}$dB, $\text{SNR}_{ij}^{\text{uses-user}} = 9$dB, $i \neq j$.



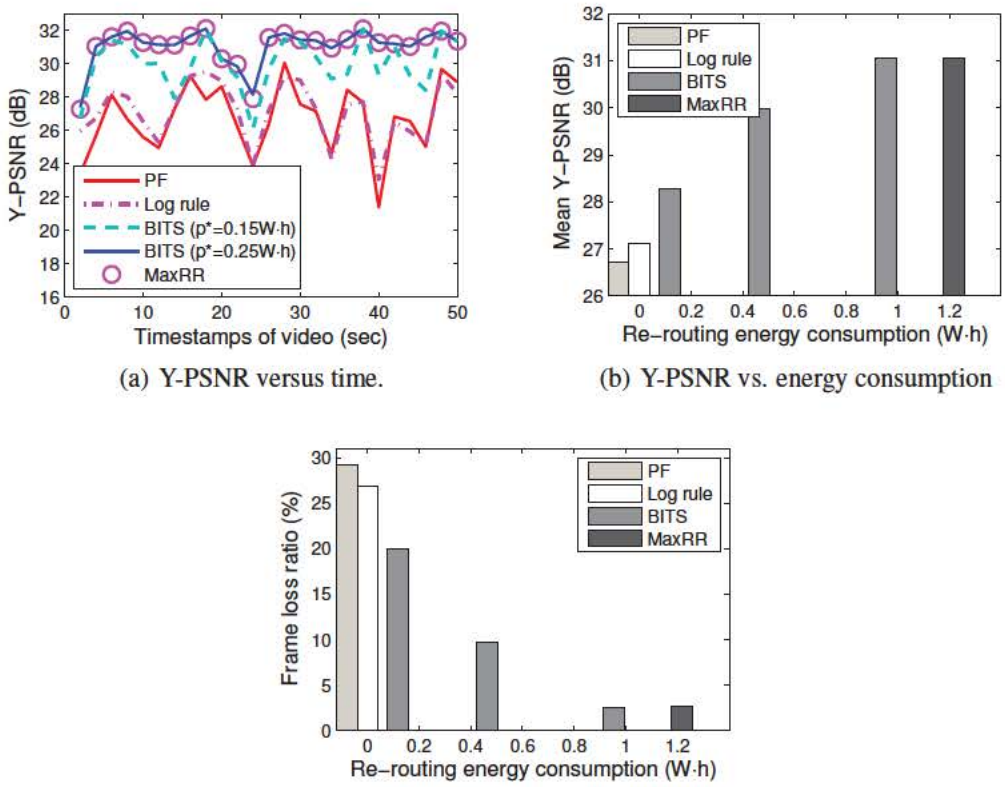

(a) Average utility (Type II) (b) Average delay (Type II)

Figure 6.9: Performance of BITS under four-user heterogeneous scenarios where $\lambda_i = 1.25 \pm 10\%$ packets/ms, $i \in \mathcal{I}$.

ence of channel conditions among users. Further, from the $5^{th}$ percentile of utility in Figure 6.9 (a) it can be observed that the utility can be increased by 1.3, and from the $95^{th}$ percentile of delay in Figure 6.9 (b) that the average packet transfer delay can be be reduced by around 55%. This demonstrate that BITS is capable of significantly improving user performance in instances where the overall system performance is poor under PF (used in current 3G and 4G systems), which is a very important practical consideration.

### 6.6.4. Simulation of live video streaming

Finally, BITS is evaluated using realistic traces of video traffic. Consider a four user homogeneous scenario, where the average $\mathbf{SNR}^{\text{bs-user}} = \{6, 6, 6, 6\}$dB and $\text{SNR}_{ij}^{\text{user-user}} = 9$dB,




(a) Y-PSNR versus time.

(b) Y-PSNR vs. energy consumption

(c) Frame loss ratio vs. energy consumption

Figure 6.10: Performance of the BITS policy in the application of live video streaming.

$\forall i, j \in \mathcal{I}$. The other wireless settings are the same as in the previous subsections. The video stream is a 50-second soccer game (4CIF YUV video with 30 frames per second) used in [104, 105]. The source YUV video is encoded into single layer H.264 Advanced Video Coding (AVC) format through the Joint Scalable Video Model (JSVM) software [105]. The encoded video is then extracted and its trace is used as the input to our simulation. The H.264 AVC video is reproduced based on the received trace, convert it to YUV format and calculate its Peak Signal-to-Noise Ratio (PSNR) with respect to the source of YUV video. Note that in the video reproduction, if the delay of a frame exceeds 200ms, it is considered lost and substitute it with the previous frame. The metrics used are the Y-PSNR of each frame and the frame loss ratio of the received video.

Results are shown in Figure 6.10. The Y-PSNR in Figure 6.10 (a) is averaged across all users over two second intervals. Figure 6.10 (a) shows that under a large energy budget, BITS can achieve performance as good as that of maximal re-routing, both of which are not sensitive to channel fluctuations most of the time (except at the beginning and several seconds around 25s where the channels between all users and the BS are bad). In contrast, the received PSNR under PF reacts greatly to the channel fluctuations, which affects the quality of experience of the users. Furthermore, it can be observed that with a lower energy budget, BITS still performs well and its

PSNR changes much more slowly than that under PF.

As for the average Y-PSNR, Figure 6.10 (b) shows that it can be improved by up to 4dB under the BITS algorithm. Similar to previous scenarios, the gain decreases with, but not as fast as, the decrease of energy budget. The frame loss ratio can be reduced greatly under BITS as shown in Figure 6.10 (c), e.g., up to 90% when compared to both PF and the log rule.

## 6.7. Summary

This chapter presented a BS-driven traffic spreading policy (BITS) to increase downlink user performance in small cells, by exploiting both BS-user and user-user communication. The problem was formulated to maximize delay-sensitive utility under a re-routing energy budget, and solved it through stochastic Lyapunov optimization. This chapter designed the BITS algorithm, studied its properties and then evaluated it in a range of scenarios. It found that BITS can greatly increase the utility and reduce the average packet transfer delay, as well as balance the additional energy consumption for traffic spreading among users. Finally, BITS was evaluated with a realistic video trace, and showed that it can increase the average Y-PSNR of the received video and reduce the frame loss ratio significantly.

# Chapter 7

# Conclusions

This thesis studied two emerging technologies in wireless networks: VLC and D2D. In Part I, I provided my research on low-cost VLC system, aiming to lower the barrier of entry to embedded VLC networks and improve the MAC protocol and link stability. In part II, I provided my research on improving the opportunistic gain in small cells through energy-aware D2D communications.

The OpenVLC platform designed, implemented, and evaluated in this thesis is the first open-source platform for VLC embedded research. It leveraged a recent powerful but cost-efficient embedded Linux platform to provide a research platform that can be used jointly with a vast array of Linux tools. It demonstrated that how a handful of commercial off-the-shelf components that could suffice as a starter kit for VLC research. Until now, OpenVLC has been used in tens of top universities in the world for research/teaching purposes, such as ETH Zurich in the Switzerland, TU Delft in the Netherlands, TU Darmstadt in Germany, Rice University, University of Wisconsin-Madison and Dartmouth College in the United States, Sungkyunkwan University in South Korea, and so on. In total, over fifty OpenVLC nodes have been deployed in the above research/teaching groups. It can be envisioned that OpenVLC be explored towards the realm of visible light by groups with no prior VLC experience, and its performance and functionalities can be improved and expanded, for instance with more sophisticated hardware and advanced PHY designs, by research groups with solid VLC backgrounds.

Beyond the OpenVLC platform, I also proposed a new MAC protocol CSMA/CD-HA MAC to enable the intra-frame bidirectional transmissions in networks of visible LEDs. This protocol adopted only one optical antenna at each node for both transmission and reception. Design, analysis, implementation, and performance evaluations of the protocol were presented. The evaluation results had showed the protocol's superior ability to detect collisions and alleviate hidden nodes, that thus boosted the system throughput.

Furthermore, motivated by the envision of the Internet of Lights, I investigated in this thesis the challenges associated with visible light links. I identified the limitations and tradeoff of two different types of optical receivers: photodiode and LED. Based on these insights, I designed and implemented a new optical data link layer that was resilient to dynamics caused by changes in

illumination and directionality. Performance evaluations had showed the proposed link outperformed traditional methods relying solely on either a photodiode or an LED. These outcomes will definitely help creating the Internet of Lights.

In the second part of this thesis, I presented the BS-transparent user-initiated traffic spreading approach to improve the downlink delay performance in small cells. This approach leveraged the latest D2D communication to recoup the opportunistic gain in small cells under the consideration of dynamic traffic load. Policy to dispatch traffic among mobile users was formulated as a Markov decision process in a two-user scenario, and heuristic algorithm was also proposed for multi-user scenarios. Simulation results had demonstrated that the proposed approach can improve the delay performance greatly. The analytical tool used there to find the optimal policy could also give hints to other researchers who wish to solve similar optimization problems.

Going forward, I proposed a BS-driven traffic spreading policy (BITS) to achieve the same goal, i.e., increasing downlink user performance in small cells, by exploiting both BS-user and D2D communication. The BITS policy was formulated to maximize delay-sensitive utility, and was solved through stochastic Lyapunov optimization. The BITS algorithm was designed and implemented in a simulator. Performance evaluations with a realistic video trace had showed that BITS could increase the quality of received video and reduce the frame loss ratio significantly.

To summarize, the research outcomes enclosed in this thesis had been published as four journal/magazine papers [52, 106–108], six conference/workshop papers [1, 69, 73, 99, 109, 110], and three demos [111–113].

# Appendices

## Appendix A

Let $b(t)$ and $s(t)$ be the stochastic process representing the backoff window size and backoff state for a given node at slot time $t$, respectively. Let $b_{i,k} = \lim_{t\to\infty}\{s(t) = i, b(t) = k\}, 0 \leq i \leq m, 0 \leq k \leq W_i - 1$ be the stationary distribution of the Markov chain, where $m$ is the maximal backoff stage and $W_i$ is the maximal backoff window size of the $i$th backoff stage. Besides, let $p$ denote the collision probability during a frame's transmission, and $\tau$ be the probability a node transmits in an arbitrary time slot. Following the analysis in [57], we can get

$$\begin{cases} p = 1 - (1-\tau)^{n-1} \\ \tau = \frac{1-p^{m+1}}{1-p} \cdot b_{0,0} \\ b_{0,0} = \frac{2(1-2p)(1-p)}{W_0(1-p)[1-(2p)^{m+1}]+(1-2p)(1-p^{m+1})} \end{cases} \tag{1}$$

Based on Eqs. (1), variables $\tau$ and $p$ can be solved numerically. Note that $\tau$ and $p$ satisfy $0 < \tau < 1$ and $0 < p < 1$.

**Saturation throughput.** Let $P_{tr}$ be the probability that the channel is busy (at least one node is transmitting) in the considered slot time, then we have

$$P_{tr} = 1 - (1-\tau)^n \tag{2}$$

The probability $P_f$ that a frame transmission is successful is given by

$$P_f = n\tau(1-\tau)^{n-1} \tag{3}$$

Moreover, the probability $P_c$ that a collision occurs during a frame's transmission is expressed as

$$P_c = P_{tr} - P_f = 1 - (1-\tau)^n - n\tau(1-\tau)^{n-1} \tag{4}$$

The time for a successful transmission $T_f$ is given by

$$T_f = T_h + E[T_p] + \delta + T_s + T_a + \delta + T_l \tag{5}$$

Let $T_c^{ca}$ and $T_c^{cd}$ be the collision time under the CSMA/CA and CSMA/CD-HA, respectively. We have

$$T_c^{ca} = T_h + E[T_p] + \delta + T_l \tag{6}$$

and

$$T_c^{cd} = T_h + E[T_p'] + \delta + T_l \tag{7}$$

Note that $E[T_p'] \ll E[T_p]$.

Let $S_{ca}$ denote the saturation throughput under CSMA/CA. Then $S_{ca}$ can be written as

$$S_{ca} = \frac{P_f E[L_p]}{(1-P_{tr})\sigma + P_f T_f + P_c T_c^{ca}} \tag{8}$$

where $\sigma$ is the duration of an empty time slot.

Similarly, let $S_{cd}$ and $S_{cd\_ha}$ be the saturation throughputs under CSMA/CD and under CSMD/CD-HA (with embedded transmission), respectively. We have

$$S_{cd} = \frac{P_f E[L_p]}{(1-P_{tr})\sigma + P_f T_f + P_c T_c^{cd}} \tag{9}$$

$$S_{cd\_ha} = \frac{P_f (E[L_p] + E[L_{em}])}{(1-P_{tr})\sigma + P_f T_f + P_c T_c^{cd}} \tag{10}$$

# Appendix B

## Proof of Lemma 1

To prove Lemma 1, we first derive the service rate $\boldsymbol{\mu(q)}$ under the delay-optimal scheduling policy (i.e. the Longest Connected Queue (LCQ) [114]). At any time $t$, a LCQ scheduler randomly chooses a queue $i$ to serve if it satisfies: *i)* Queue $i$ has the largest amount of work, where the work means the queue length divided by the service rate of the queue; *ii)* The achievable instantaneous service rate to user $i$ is positive at time $t$. Therefore, we can easily get the expression of $\boldsymbol{\mu(q)}$ under the LCQ scheduling policy:

$$\mu_i(\boldsymbol{q}) = \begin{cases} p_i, & q_i < q_j \\ (p_i + p_j - p_i p_j)/2, & q_i = q_j \\ p_i(1-p_j), & q_i > q_j \end{cases} \tag{11}$$

where $i, j \in \{1, 2\}$ and $i \neq j$. Then we make the following notations:

$$\mu_1' \equiv p_1, \quad \mu_1'' \equiv p_1(1 - p_2) \tag{12}$$

$$\mu_2' \equiv p_2(1 - p_1), \quad \mu_2'' \equiv p_2 \tag{13}$$

According to (11)-(13), we know

$$\mu_1' \geq \mu_1'', \quad \mu_2' \leq \mu_2'', \quad \mu_1' + \mu_2' = \mu_1'' + \mu_2'' \tag{14}$$

And when $p_1 = p_2$, we have

$$\mu_1' = \mu_2'', \quad \mu_1'' = \mu_2' \tag{15}$$

The uniform version of the continuous problem is present in section **??**, with uniform rate $\varphi$ and transition probabilities shown in (5.8)-(5.10). We denote $w = w_j^i, \phi = \phi_j^i, \eta = \eta_j^i, \forall i, j \in \mathcal{I}$, then Bellman's equation takes the form

$$\begin{aligned} \varphi J_{k+1}(\boldsymbol{q}) = \varphi \frac{|\,\boldsymbol{q}\,|}{|\,\boldsymbol{\lambda}\,|} + \sum_{i \in \mathcal{I}} \mu_i(\boldsymbol{q}) J_k\big([D_i \boldsymbol{q}]^+\big) \\ + \min_{\boldsymbol{\sigma}(\boldsymbol{q}) \in \mathcal{A}} \sum_{j \in \mathcal{I}} \sum_{i \in \mathcal{I}} \lambda_j \sigma_j^i(\boldsymbol{q}) \big[z_j^i \cdot (\eta + w \cdot \phi) + J_k\big(A_i \boldsymbol{q}\big)\big] \end{aligned} \tag{16}$$

*Proof*: [Poof of Lemma 1] We prove this lemma using induction. First we make the following definitions:

$$\begin{aligned} f_1(\boldsymbol{q}) \equiv & \mu_1(\boldsymbol{q} + \boldsymbol{e}_1) J_k(\boldsymbol{q}) + \mu_2(\boldsymbol{q} + \boldsymbol{e}_1) J_k([\boldsymbol{q} + \boldsymbol{e}_1 - \boldsymbol{e}_2]^+) \\ & - \mu_1(\boldsymbol{q} + \boldsymbol{e}_2) J_k([\boldsymbol{q} - \boldsymbol{e}_1 + \boldsymbol{e}_2]^+) - \mu_2(\boldsymbol{q} + \boldsymbol{e}_2) J_k(\boldsymbol{q}) \\ f_2(\boldsymbol{q}) \equiv & \lambda_1 \big\{ \min \big[J_k(\boldsymbol{q} + 2\boldsymbol{e}_1), \eta + w\phi + J_k(\boldsymbol{q} + \boldsymbol{e}_1 + \boldsymbol{e}_2)\big] \\ & - \min \big[J_k(\boldsymbol{q} + \boldsymbol{e}_1 + \boldsymbol{e}_2), \eta + w\phi + J_k(\boldsymbol{q} + 2\boldsymbol{e}_2)\big] \big\} \\ f_3(\boldsymbol{q}) \equiv & \lambda_2 \big\{ \min \big[\eta + w\phi + J_k(\boldsymbol{q} + 2\boldsymbol{e}_1), J_k(\boldsymbol{q} + \boldsymbol{e}_1 + \boldsymbol{e}_2)\big] \\ & - \min \big[\eta + w\phi + J_k(\boldsymbol{q} + \boldsymbol{e}_1 + \boldsymbol{e}_2), J_k(\boldsymbol{q} + 2\boldsymbol{e}_2)\big] \big\} \end{aligned}$$

*Induction basis:* Since $J_0(\boldsymbol{q}) = 0, \forall \boldsymbol{q} \in \mathbb{Z}^2$, we know $\Delta_0(\boldsymbol{q})$ is monotonically non-decreasing in $q_1$.

*Induction step:* Assume $\Delta_k(\boldsymbol{q})$ is monotonically non-decreasing in $q_1$ for each fixed $q_2$. By substituting (16) into (5.19), we have

$$\begin{aligned} \varphi \Delta_{k+1}(\boldsymbol{q}) = & \varphi\big(J_{k+1}(\boldsymbol{q} + \boldsymbol{e}_1) - J_{k+1}(\boldsymbol{q} + \boldsymbol{e}_2)\big) \\ = & f_1(\boldsymbol{q}) + f_2(\boldsymbol{q}) + f_3(\boldsymbol{q}) \end{aligned} \tag{17}$$

In the following we show that all $f_1(\boldsymbol{q}), f_2(\boldsymbol{q})$ and $f_3(\boldsymbol{q})$ are monotonically non-decreasing

in $q_1$ for each fixed $q_2$.

$\underline{f_1(\boldsymbol{q})}$: We have to consider different queue states around the diagonal, as shown in Fig B.1. According to different queue state, we prove this part by five different cases.

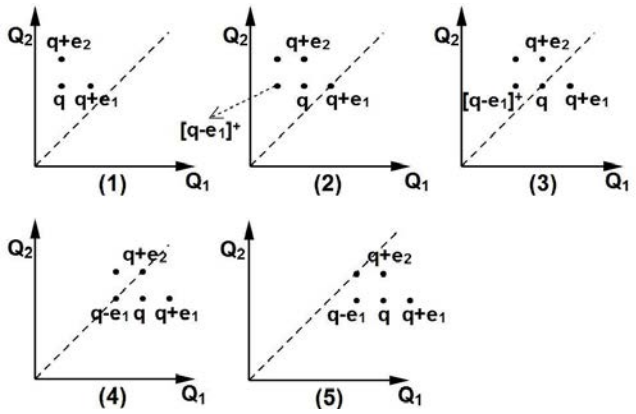



Figure B.1: Different queue states $\boldsymbol{q} \in \mathbb{Z}_+^2$ around the diagonal.

*Case 1:* If $q_1 + 1 < q_2$, according to (11)-(13) we know $\mu_1(\boldsymbol{q}+\boldsymbol{e}_1) = \mu_1(\boldsymbol{q}+\boldsymbol{e}_2) = \mu_1''$ and $\mu_2(\boldsymbol{q}+\boldsymbol{e}_1) = \mu_2(\boldsymbol{q}+\boldsymbol{e}_2) = \mu_2''$. Thus $f_2(\boldsymbol{q})$ can be written as the following:

$$\begin{aligned} f_1(\boldsymbol{q}) \equiv & \mu_1''[J_k(\boldsymbol{q}) - J_k([\boldsymbol{q}-\boldsymbol{e}_1+\boldsymbol{e}_2]^+)] \\ & + \mu_2''[J_k([\boldsymbol{q}+\boldsymbol{e}_1-\boldsymbol{e}_2]^+ - J_k(\boldsymbol{q})] \\ = & \mu_1''\Delta_k([\boldsymbol{q}-\boldsymbol{e}_1]^+) + \mu_2''\Delta_k([\boldsymbol{q}-\boldsymbol{e}_2]^+) \end{aligned} \tag{18}$$

which is non-decreasing in $q_1$ from the induction.

*Case 2:* If $q_1 + 1 = q_2$, we have $\mu_1(\boldsymbol{q}+\boldsymbol{e}_1) = \mu_1', \mu_1(\boldsymbol{q}+\boldsymbol{e}_2) = \mu_1''$ and $\mu_2(\boldsymbol{q}+\boldsymbol{e}_1) = \mu_2', \mu_2(\boldsymbol{q}+\boldsymbol{e}_2) = \mu_2''$ according to (11)-(13). Furthermore, we get $J_k(\boldsymbol{q}) = J_k(\boldsymbol{q}+\boldsymbol{e}_1-\boldsymbol{e}_2)$ since $q_1 + 1 = q_2$. Thus $f_2(\boldsymbol{q})$ can be written as:

$$\begin{aligned} f_1(\boldsymbol{q}) \equiv & \mu_1''[J_k(\boldsymbol{q}) - J_k([\boldsymbol{q}-\boldsymbol{e}_1+\boldsymbol{e}_2]^+)] \\ & + \mu_2''[J_k(\boldsymbol{q}+\boldsymbol{e}_1-\boldsymbol{e}_2) - J_k(\boldsymbol{q})] \\ = & \mu_1''\Delta_k([\boldsymbol{q}-\boldsymbol{e}_1]^+) + \mu_2''\Delta_k(\boldsymbol{q}-\boldsymbol{e}_2) \end{aligned} \tag{19}$$

which is non-decreasing in $q_1$ from the induction.

*Case 3:* If $q_1 = q_2$, we have $J_k([\boldsymbol{q}-\boldsymbol{e}_2]^+) = J_k([\boldsymbol{q}-\boldsymbol{e}_1]^+)$. Besides, we know $\mu_1(\boldsymbol{q}) =$

$\mu_2(\boldsymbol{q}) = (\mu_1' + \mu_2')/2$. Thus $f_2(\boldsymbol{q}) - f_2([\boldsymbol{q} - \boldsymbol{e}_1]^+)$ is

$$\begin{aligned}
f_1(\boldsymbol{q}) - f_1([\boldsymbol{q} - \boldsymbol{e}_1]^+) = \; & \mu_1' J_k(\boldsymbol{q}) + \mu_2' J_k([\boldsymbol{q} + \boldsymbol{e}_1 - \boldsymbol{e}_2]^+) \\
& - \mu_1'' J_k([\boldsymbol{q} - \boldsymbol{e}_1 + \boldsymbol{e}_2]^+) - \mu_2'' J_k(\boldsymbol{q}) \\
& - \frac{\mu_1'' + \mu_2''}{2} \Big\{ J_k([\boldsymbol{q} - \boldsymbol{e}_1]^+) + J_k([\boldsymbol{q} - \boldsymbol{e}_2]^+) \Big\} \\
& + \mu_1'' J_k([\boldsymbol{q} - 2\boldsymbol{e}_1 + \boldsymbol{e}_2]^+) + \mu_2'' J_k([\boldsymbol{q} - \boldsymbol{e}_1]^+) \\
= & - \big\{ \mu_1'' J_k([\boldsymbol{q} - \boldsymbol{e}_1]^+) - \mu_1'' J_k([\boldsymbol{q} - 2\boldsymbol{e}_1 + \boldsymbol{e}_2]^+) \big\} \\
= & - \Delta_k([\boldsymbol{q} - 2\boldsymbol{e}_1]^+) \geq -\Delta_k(\boldsymbol{q}) = 0
\end{aligned} \tag{20}$$

*Case 4:* The proof under this case is similar to *Case 2*.

*Case 5:* The proof under this case is similar to *Case 1*.

$\underline{f_2(\boldsymbol{q})}$: $f_2(\boldsymbol{q})$ can be written as follows:

$$\begin{aligned}
f_2(\boldsymbol{q}) = & \lambda_1 \big\{ J_k(\boldsymbol{q} + \boldsymbol{e}_1 + \boldsymbol{e}_2) - J_k(\boldsymbol{q} + \boldsymbol{e}_1 + \boldsymbol{e}_2) \\
& \quad + \min \big[ J_k(\boldsymbol{q} + 2\boldsymbol{e}_1) - J_k(\boldsymbol{q} + \boldsymbol{e}_1 + \boldsymbol{e}_2), \eta + w\phi \big] \\
& \quad - \min \big[ 0, \eta + w\phi + J_k(\boldsymbol{q} + 2\boldsymbol{e}_2) - J_k(\boldsymbol{q} + \boldsymbol{e}_1 + \boldsymbol{e}_2) \big] \big\} \\
= & \lambda_1 \big\{ \min \big[ \Delta_k(\boldsymbol{q} + \boldsymbol{e}_1), \eta + w\phi \big] \\
& + \max \big[ 0, \Delta_k(\boldsymbol{q} + \boldsymbol{e}_2) - \eta - w\phi \big] \big\} \\
= & \lambda_1 \big\{ \eta + w\phi + \min \big[ 0, \Delta_k(\boldsymbol{q} + \boldsymbol{e}_1) - \eta - w\phi \big] \\
& \qquad + \max \big[ 0, \Delta_k(\boldsymbol{q} + \boldsymbol{e}_2) - \eta - w\phi \big] \big\}
\end{aligned} \tag{21}$$

where $\Delta_k(\boldsymbol{q} + \boldsymbol{e}_1)$ and $\Delta_k(\boldsymbol{q} + \boldsymbol{e}_2)$ are monotonically non-decreasing in $q_1$ for each fixed $q_2$ from induction, resulting in $\min \big[ 0, \Delta_k(\boldsymbol{q} + \boldsymbol{e}_1) - \eta - w\phi \big]$ and $\Delta_k(\boldsymbol{q} + \boldsymbol{e}_2) - \eta - w\phi \big]$ are non-decreasing in $q_1$. Therefore, $f_2(\boldsymbol{q})$ is non-decreasing in $q_1$ for each fixed $q_2$.

$\underline{f_3(\boldsymbol{q})}$: Similarly, $f_3(\boldsymbol{q})$ is non-decreasing in $q_1$ for each fixed $q_2$, by written as follows:

$$\begin{aligned}
f_3(\boldsymbol{q}) = \lambda_2 \big\{ & - w\phi + \min \big[ 0, \Delta_k(\boldsymbol{q} + \boldsymbol{e}_1) + \eta + w\phi \big] \\
& \qquad + \max \big[ 0, \Delta_k(\boldsymbol{q} + \boldsymbol{e}_2) + \eta + w\phi \big] \big\}
\end{aligned}$$

Therefore, this lemma is proved since all $f_1(\boldsymbol{q}), f_2(\boldsymbol{q})$ and $f_3(\boldsymbol{q})$ are monotonically non-decreasing in $q_1$ for each fixed $q_2$. ■

# References


[1] D. Giustiniano and Q. Wang, "OpenVLC, an Open-Source Platform for the Internet of Light," in *Proceedings of IEEE Photonics Society Summer Topical Meeting Series*, 2015.

[2] X. Zhou and A. T. Campbell, "Visible light networking and sensing," in *Proceedings of the 1st ACM Workshop on Hot Topics in Wireless (HotWireless)*, 2014, pp. 55–60.

[3] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: A visible light based positioning system," in *Proceedings of the NSDI*, 2014, pp. 331–343.

[4] C. B. Liu, B. Sadeghi, and E. W. Knightly, "Enabling vehicular visible light communication (V2LC) networks," in *Proceedings of the ACM VANET*, 2011, pp. 41–50.

[5] H. Elgala, R. Mesleh, and H. Haas, "Indoor optical wireless communication: potential and state-of-the-art," *IEEE Communications Magazine*, vol. 49, no. 9, pp. 56–62, September 2011.

[6] L. Grobe, A. Paraskevopoulos, J. Hilt, D. Schulz, F. Lassak, F. Hartlieb, C. Kottke, V. Jungnickel, and K.-D. Langer, "High-speed visible light communication systems," *IEEE Communications Magazine*, vol. 51, no. 12, pp. 60–66, December 2013.

[7] G. Cossu, A. M. Khalid, P. Choudhury, R. Corsini, and E. Ciaramella, "3.4 Gbit/s visible optical wireless transmission based on RGB LED," *Optics Express*, vol. 20, no. 26, pp. B501–B506, December 2012.

[8] Y.-D. Lin and Y.-C. Hsu, "Multihop cellular: A new architecture for wireless communications," in *Proceedings of IEEE INFOCOM*, 2000, pp. 1273–1282.

[9] B. Kaufman and B. Aazhang, "Cellular networks with an overlaid device to device network," in *Proceedings of Asilomar Conference on Signals, Systems and Computers*, 2008, pp. 1537–1541.

[10] K. Doppler, M. Rinne, C. Wijting, C. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to LTE-advanced networks," *IEEE Communications Magazine*, vol. 47, no. 12, pp. 42–49, 2009.

[11] K. Doppler, M. P. Rinne, P. Janis, C. Ribeiro, and K. Hugl, "Device-to-device communications; functional prospects for LTE-Advanced networks," in *Proceedings of IEEE ICC Workshops*, 2009, pp. 1–6.

[12] A. Osseiran, K. Doppler, C. Ribeiro, M. Xiao, M. Skoglund, and J. Manssour, "Advances in device-to-device communications and network coding for IMT-Advanced," *ICT Mobile Summit*, 2009.

[13] T. Peng, Q. Lu, H. Wang, S. Xu, and W. Wang, "Interference avoidance mechanisms in the hybrid cellular and device-to-device systems," in *Proceedings of IEEE PIMRC*, 2009, pp. 617–621.

[14] J. Du, W. Zhu, J. Xu, Z. Li, and H. Wang, "A compressed HARQ feedback for device-to-device multicast communications," in *Proceedings of IEEE VTC-Fall*, 2012, pp. 1–5.

[15] B. Zhou, H. Hu, S.-Q. Huang, and H.-H. Chen, "Intracluster device-to-device relay algorithm with optimal resource utilization," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2315–2326, Jun. 2013.

[16] L. Lei, Z. Zhong, C. Lin, and X. Shen, "Operator controlled device-to-device communications in LTE-advanced networks," *IEEE Wireless Communications*, vol. 19, no. 3, pp. 96–104, 2012.

[17] N. Golrezaei, A. F. Molisch, and A. G. Dimakis, "Base-station assisted device-to-device communications for high-throughput wireless video networks," in *Proceedings of IEEE ICC*, 2012, pp. 7077–7081.

[18] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Device-to-device collaboration through distributed storage," in *Proceedings of IEEE GLOBECOM*, 2012, pp. 2397–2402.

[19] J. C. Li, M. Lei, and F. Gao, "Device-to-device (D2D) communication in MU-MIMO cellular networks," in *Proceedings of IEEE GLOBECOM*, 2012, pp. 3583–3587.

[20] N. K. Pratas and P. Popovski, "Low-rate machine-type communication via wireless device-to-device (D2D) links," *arXiv preprint arXiv:1305.6783*, 2013.

[21] X. Bao, U. Lee, I. Rimac, and R. R. Choudhury, "DataSpotting: offloading cellular traffic via managed device-to-device data transfer at data spots," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 14, no. 3, pp. 37–39, 2010.

[22] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushyana, and S. Viterbi, "CDMA/HDR: a bandwidth efficient high speed wireless data service for nomadic users," *IEEE Communications Magazine*, vol. 38, no. 7, pp. 70–77, Jul. 2000.

[23] X. Liu, E. K. P. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks," *Computer Networks*, vol. 41, pp. 451–474, 2003.

[24] *Cisco visual networking index: Global mobile data traffic forecast update, 2010 to 2015*. Cisco whitepaper, 2011.

[25] *Femtocell Market Status*. Femtoforum whitepaper, 2011.

[26] J. Andrews, H. Claussen, M. Dohler, S. Rangan, and M. Reed, "Femtocells: Past, present, and future," *IEEE JSAC*, vol. 30, no. 3, pp. 497–508, 2012.

[27] A. Jalali, R. Padovani, and R. Pankaj, "Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system," in *Proc. IEEE VTC 2000-Spring,*, 2000, pp. 1854–1858.

[28] "IEEE standard for local and metropolitan area networks–part 15.7: Short-range wireless optical communication using visible light," *IEEE Std 802.15.7-2011*, pp. 1–309, September 2011.

[29] T. Hao, R. Zhou, and G. Xing, "Cobra: Color barcode streaming for smartphone systems," in *Proceedings of the ACM MobiSys*, 2012, pp. 85–98.

[30] T. Yamazato, I. Takai, H. Okada, T. Fujii, T. Yendo, S. Arai, M. Andoh, T. Harada, K. Yasutomi, K. Kagawa, and S. Kawahito, "Image-sensor-based visible light communication for automotive applications," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 88–97, July 2014.

[31] S.-H. Yu, O. Shih, H.-M. Tsai, and R. Roberts, "Smart automotive lighting for vehicle safety," *IEEE Communications Magazine*, vol. 51, no. 12, pp. 50–59, 2013.

[32] N. Tippenhauer, D. Giustiniano, and S. Mangold, "Toys communicating with LEDs: Enabling toy cars interaction," in *Proceedings of the IEEE CCNC*, 2012, pp. 48–49.

[33] D. Tsonev, S. Videv, and H. Haas, "Light fidelity (Li-Fi): towards all-optical networking," in *In Proceedings of SPIE*, vol. 9007, 2013, pp. 900 702–900 702–10.

[34] H. Burchardt, N. Serafimovski, D. Tsonev, S. Videv, and H. Haas, "VLC: Beyond point-to-point communication," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 98–105, July 2014.

[35] C. Gavrincea, J. Baranda, and P. Henarejos, "Rapid prototyping of standard-compliant visible light communications system," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 80–87, July 2014.

[36] P. Dietz, W. Yerazunis, and D. Leigh, "Very low-cost sensing and communication using bidirectional LEDs," in *TR2003-35*, 2003.

[37] D. Giustiniano, N. Tippenhauer, and S. Mangold, "Low-complexity visible light networking with LED-to-LED communication," in *Proceedings of the IFIP Wireless Days (WD)*, 2012, pp. 1–8.

[38] S. Schmid, G. Corbellini, S. Mangold, and T. R. Gross, "LED-to-LED visible light communication networks," in *Proceedings of the ACM MobiHoc*, 2013, pp. 1–10.

[39] H. Chun, S. Rajbhandari, G. Faulkner, D. Tsonev, H. Haas, and D. O'Brien, "Demonstration of a Bi-directional visible light communication with an overall sum-rate of 110 Mb/s using LEDs as emitter and detector," in *IEEE Photonics Conference*, 2014, pp. 132–133.

[40] W. Popoola, E. Poves, and H. Haas, "Error performance of generalised space shift keying for indoor visible light communications," *IEEE Transactions on Communications*, vol. 61, no. 5, pp. 1968–1976, May 2013.

[41] J. H. Brown and B. Martin, "How fast is fast enough? Choosing between Xenomai and Linux for real-time applications," in *Proceedings of the Real-Time Linux Workshop*, 2010, pp. 25–27.

[42] S. Videv and H. Haas, "Practical space shift keying VLC system," in *Proceedings of the IEEE WCNC*, April 2014, pp. 405–409.

[43] Y. Park, J. Yu, J. Ko, and H. Kim, "Software radio on smartphones: Feasible?" in *Proceedings of the ACM HotMobile*, 2014, pp. 1–6.

[44] D. Bharadia, E. McMilin, and S. Katti, "Full duplex radios," in *Proceedings of the ACM SIGCOMM*, 2013, pp. 375–386.

[45] S. Wu, H. Wang, and C.-H. Youn, "Visible light communications for 5G wireless networking systems: from fixed to mobile communications," *IEEE Network*, vol. 28, no. 6, pp. 41–45, Nov 2014.

[46] W. Hu, H. Gu, and Q. Pu, "Lightsync: Unsynchronized visual communication over screen-camera links," in *Proceedings of the ACM MobiCom*, 2013, pp. 15–26.

[47] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor positioning with mobile phones and visible light," in *Proceedings of the ACM MobiCom*, 2014, pp. 447–458.

[48] Z. Wang, Z. Yang, J. Zhang, C. Huang, and Q. Zhang, "Mobile devices can afford: Lightweight indoor positioning with visible light," in *Proceedings of the ACM MobiSys*, 2015.

[49] H.-Y. Lee, H.-M. Lin, Y.-L. Wei, H.-I. Wu, H.-M. Tsai, and K. C.-J. Lin, "Rollinglight: Enabling line-of-sight light-to-camera communications," in *Proceedings of the ACM MobiSys*, 2015.

[50] A. Wang, Z. Li, C. Peng, S. Guobin, G. Fang, and B. Zeng, "Inframe++: Achieve simultaneous screen-human viewing and hidden screen-camera communication," in *Proceedings of the ACM MobiSys*, 2015.

[51] T. Li, C. An, X. Xiao, A. T. Campbell, and X. Zhou, "Real-time screen-camera communication behind any scene," in *Proceedings of the ACM MobiSys*, 2015.

[52] Q. Wang, D. Giustiniano, and D. Puccinelli, "An open-source research platform for embedded visible light networking," *IEEE Wireless Communication*, vol. 22, no. 2, pp. 94–100, 2015.

[53] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti, "Achieving single channel, full duplex wireless communication," in *Proceedings of the ACM MobiCom*, 2010, pp. 1–12.

[54] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha, "Practical, real-time, full duplex wireless," in *Proceedings of the ACM MobiCom*, 2011, pp. 301–312.

[55] K. Lin and K. Hirohashi, "High-speed full-duplex multiaccess system for LEDs based wireless communications using visible light," in *Proceedings of the International Symposium on Optical Engineering and Photonic Technology (OEPT)*, 2009, pp. 1–5.

[56] J. Zhang, X. Zhang, and G. Wu, "Dancing with light: Predictive in-frame rate selection," in *Proceedings of the IEEE INFOCOM*, 2015, pp. 1–9.

[57] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, March 2000.

[58] D. Malone, K. Duffy, and D. Leith, "Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 159–172, Feb 2007.

[59] O. Ekici and A. Yongacoglu, "IEEE 802.11a Throughput Performance with Hidden Nodes," *IEEE Communications Letters*, vol. 12, no. 6, pp. 465–467, June 2008.

[60] "BeagleBone Black," *http://beagleboard.org/Products/BeagleBone+Black*.

[61] J. Hayes and S. B. Weinstein, *Data communications principles*. Springer Science & Business Media, 1992.

[62] D. Karunatilaka, F. Zafar, V. Kalavally, and R. Parthiban, "Led based indoor visible light communications: State of the art," *Communications Surveys Tutorials, IEEE*, vol. 17, no. 3, pp. 1649–1678, thirdquarter 2015.

[63] M. Rahaim, A. Miravakili, T. Borogovac, T. Little, and V. Joyner, "Demonstration of a software defined visible light communication system," in *the 17th Annual International Conference on Mobile Computing and Networking, Mobicom2011*, 2011.

[64] Y. Qiao, H. Haas, and K. Edward, "Demo: A Software-defined Visible Light Communications System with WARP," *Demo at the ACM Workshop on Visible Light Communication Systems*, 2014.

[65] T. Li, C. An, Z. Tian, A. T. Campbell, and X. Zhou, "Human sensing using visible light communication," in *MobiCom*, 2015.

[66] C. Zhang, J. Tabor, J. Zhang, and X. Zhang, "Extending mobile interaction through near-field visible light sensing," in *MobiCom*, 2015.

[67] S. Schmid, T. Bourchas, S. Mangold, and T. R. Gross, "Linux light bulbs: Enabling internet protocol connectivity for light bulb networks," in *Proceedings of the 2nd International Workshop on Visible Light Communications Systems (VLCS)*, pp. 3–8.

[68] L. Klaver and M. Zuniga, "Shine: A Step Towards Distributed Multi-Hop Visible Light Communication," in *Proceedings of the IEEE MASS*, 2015, pp. 1–9.

[69] Q. Wang, D. Giustiniano, and D. Puccinelli, "OpenVLC: Software-Defined Visible Light Embedded Networks," in *Proceedings of the ACM Workshop on Visible Light Communication Systems (VLCS)*, 2014, pp. 1–6.

[70] P. Hu, P. H. Pathak, X. Feng, H. Fu, and P. Mohapatra, "ColorBars: Increasing Data Rate of LED-to-Camera Communication Using Color Shift Keying," in *Proceedings of the ACM CoNEXT*, 2015, pp. 1–12.

[71] N. Kashima and S. Ishii, "Optical transmission using super luminosity leds as a transmitter and a receiver," *Journal of optical communications*, vol. 23, no. 5, pp. 165–169, 2002.

[72] Y. Acharya, "Spectral and emission characteristics of led and its application to led-based sun-photometry," *Optics & Laser Technology*, vol. 37, no. 7, pp. 547–550, 2005.

[73] Q. Wang and D. Giustiniano, "Communication networks of visible light emitting diodes with intra-frame bidirectional transmission," in *Proceedings of the ACM CoNEXT*, 2014, pp. 21–28.

[74] R. Matzner and F. Englberger, "An snr estimation algorithm using fourth-order moments," in *Information Theory, 1994. Proceedings., 1994 IEEE International Symposium on*, Jun 1994, pp. 119–.

[75] S. Borst, "User-level performance of channel-aware scheduling algorithms in wireless data networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, pp. 636–647, Jun. 2005.

[76] M. Andrews, "Instability of the proportional fair scheduling algorithm for HDR," *IEEE Transactions on Wireless Communications*, vol. 3, no. 5, pp. 1422–1426, 2004.

[77] H. Falaki, D. Lymberopoulos, R. Mahajan, and etc., "A first look at traffic on smartphones," in *Proceedings of ACM IMC*, 2010, pp. 281–287.

[78] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proceedings of the USENIX*, 2010, pp. 21–21.

[79] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, and etc., "Scheduling in a queuing system with asynchronously varying service rates," *Probab. Eng. Inf. Sci.*, vol. 18, no. 2, pp. 191–217, Apr. 2004.

[80] S. Shakkottai and A. L. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The exponential rule," *American Mathematical Society Translations*, vol. 2, 2000.

[81] B. Sadiq, S. J. Baek, and G. De Veciana, "Delay-optimal opportunistic scheduling and approximations: the log rule," *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, pp. 405–418, Apr. 2011.

[82] A. Bletsas, A. Khisti, D. Reed, and A. Lippman, "A simple cooperative diversity method based on network path selection," *IEEE JSAC*, vol. 24, no. 3, pp. 659–672, 2006.

[83] S. Song, K. Son, H.-W. Lee, and S. Chong, "Opportunistic relaying in cellular network for capacity and fairness improvement," in *IEEE GLOBECOM*, 2007.

[84] H. Zhou, P. Fan, and H.-C. Yang, "Cross-layer scheduling for multiuser downlink transmissions with opportunistic relaying," in *ICCCN*, 2009.

[85] B. Zhou, S. Ma, J. Xu, and Z. Li, "Group-wise channel sensing and resource pre-allocation for LTE D2D on ISM band," in *Proceedings of IEEE WCNC*, 2013, pp. 118–122.

[86] A. Asadi and V. Mancuso, "On the compound impact of opportunistic scheduling and D2D communications in cellular networks," in *Proceedings of ACM MSWIM*, 2013.

[87] ——, "Energy efficient opportunistic uplink packet forwarding in hybrid wireless networks," in *Proceedings of ACM e-Energy*, 2013, pp. 261–262.

[88] W. Winston, "Optimality of the shortest line discipline," *Journal of Applied Probability*, vol. 14, no. 1, pp. 181–189, 1977.

[89] A. Ephremides, P. Varaiya, and J. Walrand, "A simple dynamic routing problem," *IEEE Transactions on Automatic Control*, vol. 25, no. 4, pp. 690 – 693, Aug. 1980.

[90] B. Hajek, "Optimal control of two interacting service stations," *IEEE Transactions on Automatic Control*, vol. 29, pp. 491–499, Jun. 1984.

[91] IEEE Std 802.11n, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput," pp. 1–565, 2009.

[92] R. Prakash and V. V. Veeravalli, "Centralized wireless data networks with user arrivals and departures," *IEEE Transactions on Information Theory*, vol. 53, no. 2, pp. 695–713, 2007.

[93] D. P. Bertsekas, *Dynamic Programming and Optimal Control (Volume I)*. Athena Scientific, 2005.

[94] T. M. Farrar, "Optimal use of an extra server in a two station tandem queueing network," *IEEE Transactions on Automatic Control*, vol. 38, no. 8, pp. 1296–1299, 1993.

[95] J. Sommers and P. Barford, "Cell vs. WiFi: on the performance of metro area mobile connections," in *Proceedings of ACM IMC*, 2012, pp. 301–314.

[96] B. Sadiq, S. J. Baek, and G. de Veciana, "Delay-optimal opportunistic scheduling and approximations: the log rule," in *IEEE INFOCOM*, 2009.

[97] P. Liu, R. Berry, and M. Honig, "Delay-sensitive packet scheduling in wireless networks," in *IEEE WCNC*, 2003.

[98] G. Song, Y. Li, L. Cimini, and H. Zheng, "Joint channel-aware and queue-aware data scheduling in multiple shared wireless channels," in *IEEE WCNC*, 2004.

[99] Q. Wang and B. Rengarajan, "Recouping opportunistic gain in dense base station layouts through energy-aware user cooperation," in *Proceedings of IEEE WoWMoM*, 2013, pp. 1–9.

[100] R. Ganti and M. Haenggi, "Spatial analysis of opportunistic downlink relaying in a two-hop cellular system," *IEEE Transactions on Communications*, vol. 60, no. 5, pp. 1443–1450, 2012.

[101] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.

[102] S. Sesia, I. Toufik, and M. Baker, *LTE: the UMTS long term evolution*. Wiley, 2011.

[103] T.-S. Kim, H. Lim, and J. C. Hou, "Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks," in *ACM Mobicom*, 2006.

[104] A. Detti, G. Bianchi, C. Pisa, F. Proto, P. Loreti, W. Kellerer, S. Thakolsri, and J. Widmer, "SVEF: an open-source experimental evaluation framework," in *IEEE MediaWIN*, 2009.

[105] JSVM. http://www.hhi.fraunhofer.de/de/kompetenzfelder/image-processing/research-groups/image-video-coding/svc-extension-of-h264avc/jsvm-reference-software.html.

[106] Q. Wang and D. Giustiniano, "Intra-Frame Bidirectional Transmission in Networks of Visible LEDs," *Accepted by IEEE/ACM Transactions on Networking*, pp. 1–13, 2016.

[107] Q. Wang, B. Rengarajan, and J. Widmer, "Increasing opportunistic gain in small cells through energy-aware user cooperation," *IEEE Transactions on Wireless Communications*, vol. 13, no. 11, pp. 6356–6369, Nov 2014.

[108] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.

[109] Q. Wang, B. Rengarajan, and J. Widmer, "Increasing opportunistic gain in small cells through base station-driven traffic spreading," in *Proceedings of IEEE WoWMoM*, 2014, pp. 1–9.

[110] Q. Wang, D. Giustiniano, and O. Gnawali, "Low-cost, flexible and open platform for visible light communication networks," in *Proceedings of the 2nd International Workshop on Hot Topics in Wireless (HotWireless)*, 2015, pp. 31–35.

[111] Q. Wang, S. Yin, O. Gnawali, and D. Giustiniano, "Demo: OpenVLC1.0 Platform for Research in Visible Light Communication Networks," in *Proceedings of the ACM MobiCom*, 2015, pp. 179–181.

[112] Q. Wang, D. De Donno, and D. Giustiniano, "Demo: OpenVLC1.1 - Open-Source Research Platform for the Internet of Lights," in *Proceedings of the ACM IPSN*, 2016, pp. 1–2.

[113] Q. Wang, D. Giustiniano, and D. Puccinelli, "OpenVLC: Software-Defined Open Architecture for Embedded Visible Light Networks," in *(Demo) the ACM Workshop on Visible Light Communication Systems (VLCS)*, 2014, pp. 1–2.

[114] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Transactions on Information Theory*, vol. 39, no. 2, pp. 466–478, Mar. 1993.